



**Rafael Lopes Mendonça      Sistema de informação para doenças e mutações  
genéticas**



**Rafael Lopes Mendonça**

**Sistema de informação para doenças e mutações  
genéticas**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia de Computadores e Telemática, realizada sob a orientação científica do Dr. José Luís Oliveira, Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

Dedico este trabalho à minha família e amigos.



## **o júri**

presidente

**Professor Doutor Armando José Formoso de Pinho**

Professor Associado com Agregação do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro

**Professor Doutor António Manuel de Jesus Pereira**

Professor Coordenador do Departamento de Engenharia Informática da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria

**Professor Doutor José Luís Guimarães Oliveira**

Professor Associado do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro



## **agradecimentos**

Um obrigado a todos os colegas do grupo de Bioinformática, em especial ao Professor Doutor José Luís Oliveira, meu orientador de tese, pela oportunidade que me deu em efectuar este trabalho e ter acreditado em mim, e ao Pedro Lopes pelo importante apoio.

Um especial agradecimento a todos os meus amigos, pelo apoio que sempre demonstraram, pelos bons momentos e sua amizade.

À namorada, pela compreensão e paciência que teve, e por nunca ter deixado de me ajudar sempre que precisei.

Por fim um grande e sentido “Muito Obrigado” à minha família, em especial aos meus pais e irmã, pelo apoio incondicional, paciência e força que me deram ao longo do curso, muito importante em momentos difíceis.

A todos, um sincero obrigado.





**palavras-chave**

Programação Web, Doenças raras, Genes, Mutações

**resumo**

A informática está cada vez mais presente no apoio à medicina, surgindo como uma importante ferramenta, quer no acompanhamento clínico, quer na partilha de conhecimento, de forma a aumentar a percentagem de sucesso no diagnóstico e posteriormente, na fase do tratamento das doenças.

No grupo das doenças encontra-se um subgrupo a que se dá o nome de raras ou órfãs devido ao número reduzido de pessoas que sofrem deste tipo de doenças. São por vezes incapacitantes, reduzindo a qualidade de vida e afectam os familiares que os rodeiam.

O presente trabalho tem como objectivo a construção de um sistema de suporte à prática e à investigação clínica, no âmbito das doenças raras, de forma a facilitar a recolha de informação, diagnóstico, tratamento e apoio aos pacientes. Através deste sistema poder-se-á otimizar os recursos médicos especializados, os recursos informáticos e a qualidade dos dados, para que, posteriormente, essa informação esteja disponível para consulta e apoio em futuras decisões.

O sistema desenvolvido permite igualmente que os profissionais de saúde tenham disponível uma plataforma Web para partilharem esta informação, importante para o incremento na qualidade e avanço tecnológico nesta área da saúde.



**keywords**

Web Programming, Rare Diseases, Genes, Mutations

**abstract**

Information technology is increasingly present in medicine, and it is emerging as a crucial tool both in clinical monitoring or knowledge dissemination, increasing the success rate in the diagnosis and subsequently during treatments.

A particular sub-group of diseases are called rare or orphan due to the small number of people suffering from these illnesses. They are sometimes disabling, reducing the quality of life of patients and affect all relatives around them.

The present work aims to build a Web system to support the practice and clinical research in rare diseases, and to facilitate the collection of information, diagnosis, treatment and patient support. Through this system one will be able to optimize specialized medical resources, computing resources and data quality, so that, later, that information is available for consultation and for supporting future decisions.

The developed system also allows health professionals to share information that will be important for enhancing the quality and technological advancement in this area of health.



# Índice

<b>LISTA DE FIGURAS .....</b>	<b>I</b>
<b>LISTA DE TABELAS .....</b>	<b>III</b>
<b>LISTA DE ACRÓNIMOS.....</b>	<b>IV</b>
<b>1. INTRODUÇÃO .....</b>	<b>1</b>
1.1. ENQUADRAMENTO .....	1
1.2. OBJECTIVOS.....	2
1.3. ESTRUTURA DA DISSERTAÇÃO.....	2
<b>2. ESTADO DE ARTE.....</b>	<b>5</b>
2.1. DOENÇAS RARAS .....	5
2.1.1. Organizações e Projectos .....	5
2.1.2. Informação Científica .....	7
2.1.3. LSDB (Locus-Specific Database) .....	7
2.2. TECNOLOGIAS DE PROGRAMAÇÃO PARA WEB.....	9
2.2.1. Model-View-Controller.....	9
2.2.2. Stripes .....	11
2.2.3. GWT .....	12
2.2.4. Struts .....	13
2.2.5. Struts 2.....	14

2.2.6.	JSF .....	15
2.2.7.	Análise comparativa .....	15
2.2.8.	Outras tecnologias utilizadas .....	17
2.2.9.	Persistência de dados .....	18
2.3.	SUMÁRIO .....	20
<b>3.</b>	<b>ANÁLISE DE REQUISITOS .....</b>	<b>21</b>
3.1.	VISÃO GERAL DO SISTEMA .....	21
3.2.	CASOS DE UTILIZAÇÃO .....	23
3.2.1.	Actores .....	23
3.2.2.	Administração do sistema .....	24
3.2.3.	Administração do centro de referência .....	26
3.3.	MÓDULOS DE TRABALHO .....	28
3.4.	REQUISITOS NÃO FUNCIONAIS .....	30
3.5.	SUMÁRIO .....	30
<b>4.</b>	<b>DESENVOLVIMENTO DO SISTEMA DE GESTÃO DE DOENÇAS RARAS .....</b>	<b>31</b>
4.1.	ARQUITECTURA .....	31
4.1.1.	Modelo Físico .....	32
4.2.	MODELO DE DADOS .....	33
4.2.1.	Centro de Referência .....	34
4.2.2.	Informação para as Doenças Raras .....	35
4.2.3.	Formulário Dinâmico .....	36
4.2.4.	Estudo .....	37
4.3.	FUNCIONALIDADES .....	39
4.3.1.	Módulos da Aplicação .....	39
4.3.2.	Autenticação .....	41
4.3.3.	Criar Centro de Referência .....	42
4.3.4.	Fase de Configuração do Centro de Referência .....	42
4.3.5.	Formulários Dinâmicos .....	43

4.3.6. Recolha de Informação das Doenças Raras.....	45
4.3.7. Gestão dos Estudos .....	47
4.3.8. Registo de acções.....	49
4.4. INTERACÇÃO E INTEGRAÇÃO COM O LOVD.....	49
4.4.1. Criar uma Instância LOVD .....	49
4.4.2. Gestão das Instâncias LOVD .....	51
4.4.3. Autenticação no LOVD.....	52
4.4.4. Procedimento para Adicionar um Paciente.....	53
4.4.5. Recolha de Informação de Mutações Genéticas Relativas a um Paciente .....	53
4.5. SUMÁRIO.....	54
<b>5. RESULTADOS .....</b>	<b>55</b>
5.1. LOGIN E AUTENTICAÇÃO .....	55
5.2. CENTRO DE REFERÊNCIA .....	56
5.2.1. Configuração Inicial .....	57
5.2.2. Gestão dos Membros .....	59
5.2.3. Gestão dos Formulários Dinâmicos.....	60
5.2.4. Gestão dos Pacientes .....	61
5.2.5. Recolha e Registo de Informação de Doenças Raras .....	63
5.2.6. Utilização do LOVD .....	64
5.3. SUMÁRIO.....	64
<b>6. CONCLUSÃO .....</b>	<b>65</b>
6.1. SUGESTÕES PARA TRABALHOS FUTUROS.....	65
<b>7. REFERÊNCIAS.....</b>	<b>67</b>

# Lista de Figuras

FIGURA 1 - ARQUITECTURA MVC.....	10
FIGURA 2 - FLUXO TÍPICO DO STRIPES . ....	11
FIGURA 3 - PROCESSO DE IMPLEMENTAÇÃO DE UMA APLICAÇÃO GWT.....	12
FIGURA 4 - FLUXO TÍPICO DO STRUTS.....	14
FIGURA 5 - FLUXO TÍPICO DO STRUTS 2. ....	14
FIGURA 6 - TOPO DA ARQUITECTURA JSF.....	15
FIGURA 7 - ARQUITECTURA DO MECANISMO DE ARMAZENAMENTO DO MYSQL . ....	19
FIGURA 8 - PROCESSO DO ACOMPANHAMENTO CLÍNICO EFECTUADO NUM PACIENTE.....	22
FIGURA 9 - MODELO DE DADOS DOS UTILIZADORES. ....	24
FIGURA 10 - DIAGRAMA DOS CASOS DE UTILIZAÇÃO DO ADMINISTRADOR.....	25
FIGURA 11 - DIAGRAMA DOS CASOS DE UTILIZAÇÃO DO MEMBRO. ....	27
FIGURA 12 - DIAGRAMA DE ACTIVIDADE DE UM NOVO ESTUDO. ....	28
FIGURA 13 - DIAGRAMA DE ACTIVIDADES NUM CENTRO DE REFERÊNCIA .....	29
FIGURA 14 - ARQUITECTURA GERAL DO SISTEMA.....	32
FIGURA 15 - MODELO FÍSICO DO SISTEMA. ....	33
FIGURA 16 - DIAGRAMA DE CLASSES CENTRALIZADO NO CENTRO DE REFERÊNCIA. ....	35
FIGURA 17 - DIAGRAMA DE CLASSES CENTRADO NA DOENÇA RARA. ....	36
FIGURA 18 - DIAGRAMA DE CLASSES CENTRALIZADO NO FORMULÁRIO.....	37
FIGURA 19 - DIAGRAMA DE CLASSES CENTRALIZADO NO ESTUDO.....	38
FIGURA 20 - DIAGRAMA DE ESTADOS DA AUTENTICAÇÃO NO SISTEMA. ....	41
FIGURA 21 - FORMULÁRIO PARA ADICIONAR UM NOVO CENTRO DE REFERÊNCIA. ....	42
FIGURA 22 - FASES DE SETUP DE CONSTRUÇÃO DE UM CENTRO DE REFERÊNCIA .....	43
FIGURA 23 - ARQUITECTURA DA CONSTRUÇÃO DE UM FORMULÁRIO.....	44
FIGURA 24 - MODELO DE DADOS DA INFORMAÇÃO DE UMA DOENÇA RARA. ....	45
FIGURA 25 - DIAGRAMA DE SEQUÊNCIA DO REGISTO DOS GENES NO SISTEMA E NO LOVD. ...	47
FIGURA 26 - MODELO DE INFORMAÇÃO DE UM PACIENTE.....	48
FIGURA 27 - FASES NA CONSTRUÇÃO DE UMA INSTÂNCIA LOVD.....	51
FIGURA 28 - DIAGRAMA DE SEQUÊNCIA DE ACESSO À INTERFACE LOVD. ....	52
FIGURA 29 - PÁGINA DE AUTENTICAÇÃO.....	55



FIGURA 30 - EXEMPLIFICAÇÃO DA INSERÇÃO DE DADOS NUM LOGIN INCORRECTOS. ....	56
FIGURA 31 - INTERFACE CLIENTE APÓS VALIDAÇÃO DO UTILIZADOR. ....	56
FIGURA 32 - EXEMPLO DA PÁGINA INICIAL DE UM CENTRO DE REFERÊNCIA.....	57
FIGURA 33 - CONFIGURAÇÃO INICIAL DE UM CENTRO DE REFERÊNCIA. ....	57
FIGURA 34 - VISUALIZAÇÃO DA LISTA DE DOENÇAS RARAS.....	58
FIGURA 35 - EXEMPLO DA UTILIZAÇÃO DO <i>AUTOCOMPLETE</i> . ....	58
FIGURA 36 - INTERFACE PARA CRIAR UM NOVO FORMULÁRIO. ....	59
FIGURA 37 - LISTA DE MEMBROS QUE FAZEM PARTE DO CENTRO DE REFERÊNCIA.....	60
FIGURA 38 - INTERFACE PARA ADICIONAR UM NOVO MEMBRO. ....	60
FIGURA 39 - LISTA DE FORMULÁRIOS DISPONÍVEIS PARA UTILIZAÇÃO NOS ESTUDOS.....	61
FIGURA 40 - LISTA DE PACIENTES PRESENTES NA APLICAÇÃO.....	61
FIGURA 41 - FORMULÁRIO PARA O REGISTO DE UM NOVO PACIENTE.....	62
FIGURA 42 - INTERFACE DE DADOS RELATIVO A UM PACIENTE. ....	62
FIGURA 43 - INTERFACE DE UM ESTUDO. ....	63
FIGURA 44 - LISTA DE DOENÇAS RARAS INCLUÍDAS NO CENTRO DE REFERÊNCIA.....	63
FIGURA 45 - INTERFACE LOVD UTILIZADA PELO SISTEMA. ....	64

# Lista de Tabelas

TABELA 1 - COMPARAÇÃO ENTRE AS <i>FRAMEWORKS</i> .....	16
TABELA 2 - ACTORES DO SISTEMA. ....	24
TABELA 3 - OUTROS CASOS DE UTILIZAÇÃO.....	26

# Lista de Acrónimos

ADN	Ácido Desoxirribonucleico
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CSV	Comma-separated values
DOM	Document Object Model
EURORDIS	Europe Rare Diseases
HGNC	HUGO Gene Nomenclature
HGVS	Human Genome Variation Society
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
JPA	Java Persistence API
LSDB	Locus Specific Database
LOVD	Leiden Open Variation Database
MVC	Model-View-Controller
OMIM	Online Mendelian Inheritance in Man
SNS	Sistema Nacional de Saúde

# I. Introdução

## 1.1. Enquadramento

Em Portugal calcula-se que existam aproximadamente entre 5000 a 8000 doenças raras, que afectam cerca de 6% a 8% da população. Destas destacam-se as de origem genética surgindo em aproximadamente 80% dos casos [1]. Este tipo de doenças, devido ao número muito reduzido de pacientes, têm dificuldades em encontrar apoio, quer a nível clínico, quer a nível psicológico [2]. Algumas destas doenças são crónicas, afectam gravemente a qualidade de vida dos pacientes e causam sérias incapacidades e danos em termos sociais, principalmente, nos familiares que os rodeiam. Assim, torna-se fundamental um acompanhamento médico especializado de modo a potencializar os mecanismos de apoio, tratamento e investigação neste tipo de doenças.

O facto de existirem um número reduzido de pacientes por cada doença rara, prejudica a criação de um número mínimo de casos necessários para estudos adequados. Neste sentido, as novas tecnologias aparecem como um importante complemento ao dispor destes profissionais de saúde, através de ferramentas de apoio clínico, permitindo melhorar os procedimentos e recursos médicos, que muitas vezes são escassos, de forma a ser o mais preciso e correcto possível. Um correcto diagnóstico precoce e um constante acompanhamento são decisivos para atenuar e melhorar a qualidade de vida dessas pessoas, e se possível atingir a cura.

Com este intuito surgiram os denominados centros de referência, que são estruturas organizativas constituídas por equipas multidisciplinares que efectivam o apoio a esses pacientes. Associada a estes recursos humanos, emerge a necessidade de construir uma plataforma Web para implementar esse acompanhamento, integrando toda a informação obtida pelos estudos de forma a existir um banco de informação sobre essas doenças raras e seus pacientes.

## 1.2. Objectivos

Este trabalho tem como objectivo principal desenvolver uma aplicação Web que integre as ferramentas necessárias para colocar em prática a formação de uma rede de centros de referência no âmbito das doenças raras. Pretende-se com esta plataforma armazenar os dados recolhidos pelas diferentes entidades, através dos seus clínicos especializados, permitindo assim uma clara organização, no que respeita aos recursos humanos e a informação sobre este tipo de doenças.

Como para cada doença rara existem dados específicos que podem variar a sua estrutura ao longo do tempo, é necessário desenvolver os instrumentos necessários para uma caracterização completa, com a implementação de um sistema de formulários dinâmicos, que permita facilitar a utilização por parte dos clínicos quando efectuam os registos dos estudos.

Através de mais alguns objectivos específicos, foi possível detalhar as seguintes funcionalidades que se pretende obter durante este trabalho:

- Configuração de centros de referências
- Gestão de centros de referências
- Registo único do paciente
- Registo clínico do paciente, relativamente a doenças raras
- Registo de mutações genéticas
- Partilha de informação entre os membros de cada centro de referência

## 1.3. Estrutura da Dissertação

Após este primeiro capítulo introdutório, a dissertação apresenta a seguinte estrutura:

- No segundo capítulo é apresentado o estado de arte no âmbito das doenças raras, bem como os projectos e organizações que existem nesta área. É ainda efectuada uma avaliação de ferramentas utilizadas para o desenvolvimento da plataforma.

- O terceiro capítulo apresenta os requisitos que esteve na base da implementação da aplicação, explicando detalhadamente as principais funcionalidades que o sistema deve conter.
- O quarto capítulo centra-se na apresentação da aplicação. É descrito a arquitectura do sistema, os procedimentos e a estratégia adoptada para a implementação da plataforma.
- No quinto capítulo são apresentados os resultados do sistema, com a descrição das interfaces e funcionalidades do portal, sendo demonstrado um protótipo da solução.
- Por fim, o sexto capítulo é dedicado às conclusões, bem como propostas futuras que poderão ser efectuados neste sistema apresentado.



## 2. Estado de Arte

Neste capítulo será efectuada inicialmente uma análise ao que tem vindo a ser desenvolvido na área das doenças raras. De seguida, é apresentada uma análise de algumas plataformas existentes para o desenvolvimento de aplicações Web, sendo depois justificada a escolha da solução usada tendo em conta critérios que se adequam a este trabalho.

### 2.1. Doenças Raras

O genoma humano divide-se em 23 pares de cromossomas que contêm internamente um conjunto de genes. Estes genes são susceptíveis de conservar mutações transmitidas pela família ou adquiridas ao longo da vida. A descodificação do genoma humano tem vindo a permitir à comunidade científica conhecer um pouco mais sobre as causas da maioria das doenças raras [3].

Devido a grande parte das doenças raras terem origem genética, ou seja, são encontradas anomalias na estrutura genética, é crucial a recolha e salvaguarda desta informação, identificando os genes responsáveis pela afectação destas doenças. Estes genes apresentam factores de risco podendo apresentar variações, que ao serem identificadas serão úteis como objecto de estudo e investigação.

Neste tipo de doenças raras é essencial efectuar testes genéticos o mais rapidamente possível, para efectuar uma despistagem e, em casos positivos, poder realizar uma terapêutica de forma a melhorar a qualidade de vida destes pacientes [3, 4].

#### 2.1.1. Organizações e Projectos

São vários os projectos e organizações que se têm dedicado ao desenvolvimento de soluções médicas para determinadas doenças raras, e que têm contribuído para um incremento na qualidade dos serviços prestados nesta área.



A nível Europeu destaca-se a EURORDIS (Europe Rare Diseases) [5], formada por associações de pacientes e pessoas que se dedicam a melhorar a qualidade de vida de quem vive com uma doença rara. Tem como principal objectivo desenvolver uma plataforma com os registos de pessoas com este tipo de doenças na União Europeia e pretende actuar em questões regulamentares, étnicas, técnicas e financeiras [6].

Entretanto, na União Europeia surgiu a necessidade da formação da RDTF (Rare Disease Task Force) [7], com a missão de aconselhar e assistir a Comissão Europeia Direcção de Saúde Pública na promoção do ideal de prevenção, diagnóstico e tratamento das doenças raras na Europa e proporcionar um fórum de discussão com troca de opiniões e experiências sobre todos os assuntos relacionados com este tipo de doenças.

Marco importante tem sido ORPHANET [2], uma plataforma Web direccionada ao público em geral, profissionais de saúde e pacientes, dedicada à informação sobre medicamentos órfãos e doenças raras. Apresenta informações sobre consultas especializadas, laboratórios de diagnóstico, projectos de investigação, ensaios clínicos, e grupos de apoio.

Outro dos projectos que surgiram foi o E-Rare, que pretende ser uma ferramenta para superar as diferenças de cada país, desenvolvendo um programa comum de investigação. No seguimento deste projecto surgiu uma segunda fase, denominada E-Rare2, alargando e aprofundando o conhecimento sobre este tipo de doenças em todo o mundo e financiando as actividades de investigação, ampliando a cooperação e investigação [8].

Destaca-se também a organização NORD (National Organization for Rare Disorders) [9], cujo objectivo prende-se com a identificação, tratamento e a cura destes pacientes, através de programas de educação, jurídica, serviços e investigação, contendo também uma base de dados com informações específicas e detalhadas sobre estas doenças.

Em forma de nota conclusiva, tem-se verificado um aumento no interesse em resolver as insuficiências nesta área da saúde, com o incremento no investimento de projectos direccionados para evolução e investigação neste âmbito das doenças raras.

## 2.1.2. Informação Científica

Para a comunicação científica é fundamental a utilização de termos precisos que transmitam de forma clara e concisa a informação médica. Para isso, existem convenções de nomenclatura que, quando conhecidas e utilizadas por todos, facilitam o entendimento entre os profissionais da área. Neste trabalho, destacam-se a OMIM (*Online Mendelian Inheritance in Man*) [10] e HGNC (*HUGO Gene Nomenclature*) [11].

OMIM é uma base de dados online com informação dos genes humanos e pretende descrever detalhadamente as doenças genéticas. Cada doença e gene tem associado um código de identificação que facilita a pesquisa genética humana, na educação e sua utilização na investigação de genética clínica. Cada entrada OMIM tem informação de um fenótipo, contendo várias ligações para outras bases de dados de informação genética. Tem ainda disponível para download ficheiros *morbidmap* relativos a todas as doenças raras conhecidas e o *genemap* relativo aos genes. A cada entrada OMIM é atribuído um número identificador único, com seis dígitos [10].

O HGNC é uma entidade dedicada à coordenação de esforços colaborativos na pesquisa sobre o genoma humano [11-13]. Tem como objectivo aprovar um nome e um símbolo para cada gene catalogado, seguindo certas regras que visam facilitar tanto a comunicação entre investigadores como o armazenamento e pesquisa nas bases de dados. Cada gene recebe um nome que deve ser curto e contém informações sobre sua função e sua relação com outros genes do mesmo grupo. Já existem mais de 30000 genes catalogados na base de dados do HGNC, além do nome e símbolo oficiais, informa também a localização do cromossoma e outros símbolos associados, bem como os sinónimos de cada gene. Além dessas informações, fornece ligações para diversas bases de dados existentes que caracterizam os genes como OMIM, GenBank, Ensembl, e UniProt.

## 2.1.3. LSDB (*Locus-Specific Database*)

Uma LSDB é uma base de dados que contém dados de mutações e centra-se na informação sobre variações num único gene, e geralmente utilizada por um conjunto de investigadores de uma forma colaborativa, com especialidade num determinado gene ou fenótipo. Fornece uma ferramenta valiosa para a análise da expressão

genética e fenótipo, tanto em condições normais quer em doença. Este sistema promove a apresentação da informação e mantém uma precisão e actualização da fonte de dados, importante para investigação e pesquisa [14].

Existem algumas soluções para construção de uma LSDB. Os mais conhecidos são:

- MutaBase [15]
- UMD [16]
- LOVD [17]

Tendo em conta a necessidade de recorrer a uma LSDB para o registo das mutações genéticas neste trabalho, foi escolhido o sistema LOVD por ser uma solução que pode ser instalada localmente e pela sua utilização por grande parte dos profissionais nesta área.

LOVD (Leiden Open Variation Database) é então uma LSDB, direccionada ao gene, que permite o registo das mutações genéticas, associando-as a pacientes em que foram diagnosticadas. É importante incluir esta aplicação no presente trabalho porque vai permitir a recolha de informação sobre os genes e as suas variações, com o propósito de efectuar o registo e visualização das mutações genéticas no ADN. É uma plataforma web-based disponível em código aberto, e utiliza o software PHP como linguagem de programação base, em que o sistema de gestão da base de dados é o MySQL. A sua base de dados segue as recomendações HGVS (Human Genome Variation Society), deste modo permite que a nomenclatura utilizada seja generalizada por todos os utilizadores.

Para a instalação necessita de um servidor Web, como o Apache, e requer uma configuração prévia aquando da sua instalação. Para se efectuar uma correcta instalação do LOVD é necessário o registo do estado actual da aplicação (tabela *status*), o registo da configuração (tabela *config*), e adicionar o utilizador inicial responsável, o administrador (tabela *user*). Quando é adicionado um gene para estudo no LOVD é necessário criar duas tabelas, ou seja, por cada entrada de um gene referenciado (tabela *genes*), são necessárias duas tabelas adicionais para o registo das variantes (*gene\_columns*, *gene\_variants*).

O LOVD contém cinco níveis de acesso: *Administrator*, *manager*, *curator*, *users* e os *submitters*. Permite também, a edição de colunas que persistem na base dados, tanto

ao nível das variações dos genes como relativamente aos pacientes, útil para se proceder à modelação de acordo com o sistema que pretendemos neste trabalho.

O LOVD permite ainda que sejam desenvolvidos módulos de forma a complementar a plataforma de acordo com o desejado para a sua utilização. Quando da instalação, contém três módulos disponíveis que podem ser instalados:

- **Mutalyzer:** Efectua a verificação da nomenclatura da sequência da variação de acordo com os guias do HGVS.
- **ShowMaxDBID:** Modifica a forma de apresentação da mutação genética.
- **reCAPTCHA:** Altera o formulário de inscrição dos *submitters* de forma a proteger contra a falsificação de registos.

## 2.2. Tecnologias de programação para Web

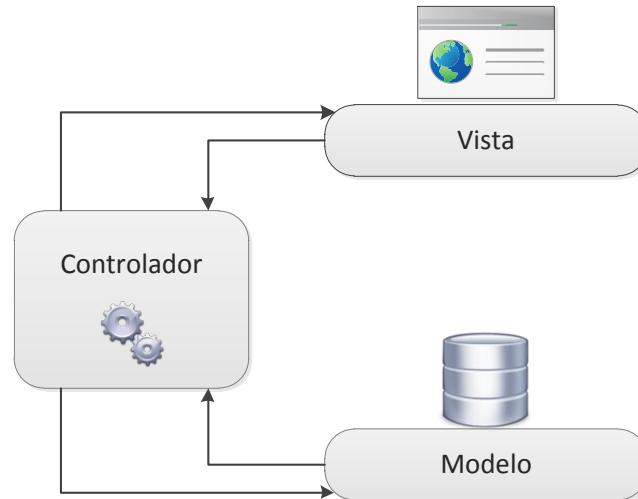
Neste tópico, vai ser efectuada uma comparação e a justificação na escolha das tecnologias usadas, que têm como base a linguagem de programação em Java, escolhida para a fase de desenvolvimento.

### 2.2.1. *Model-View-Controller*

A estratégia de programação aplicada no sistema a implementar segue uma arquitectura *Model-View-Controller*. Esta organização divide a aplicação em três segmentos: a) o modelo, que representa os elementos do domínio e interacção com as ferramentas de persistência; b) a vista, que representa a interface com o utilizador da plataforma; c) e o controlo, que processa e responde a eventos gerados pelo utilizador e interage com o modelo.

O principal objectivo desta arquitectura é permitir efectuar alterações nos diferentes segmentos sem que haja necessidade de alterar a estrutura da aplicação. Também

podemos tirar partido da possibilidade de que a interacção com a aplicação seja efectuada por diferentes interfaces [18]. Deste modo, uma possível representação da arquitectura MVC pode ser dada pela Figura 1.



**Figura 1** - Arquitectura MVC.

De acordo com este modelo são descritas as seguintes camadas:

- **Controlador:** Efectua a gestão de eventos, geralmente acções do utilizador, e pode invocar alterações no Modelo. É aqui que é efectuada a validação dos dados e também é onde os valores introduzidos pelos utilizadores são filtrados. Efectua o controlo do Modelo e faz a troca apropriada na vista. O controlador aceita entradas e dá instruções para que o modelo e a apresentação actuem em função dessas entradas.
- **Modelo:** Representa os dados e as funcionalidades da aplicação. Os dados não têm que ter necessariamente um mecanismo de armazenamento persistente para guardar os dados, embora não seja o caso no presente sistema.
- **Vista:** Este nível permite visualizar o estado do modelo, e interagir, geralmente atrás da interface com o utilizador. É efectuada a gestão da apresentação através de gráficos ou texto.

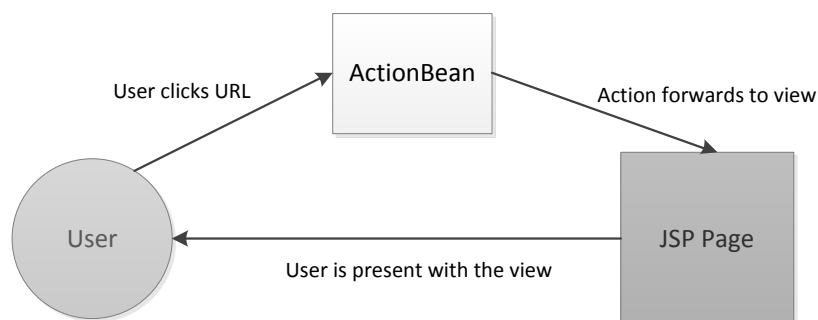
Nas aplicações Web esta arquitectura permite a separação das camadas pelas diferentes máquinas. Do lado do cliente encontra-se o browser com acesso à interface de cliente da aplicação apenas com a utilização da vista e controlo, estendendo-se o controlo até ao lado servidor onde se encontram a lógica de dados e o modelo da aplicação. Deste modo o controlo actua tanto do lado cliente como do lado servidor, sendo responsável por transformar os eventos do utilizador em serviços que serão processados no lado servidor e, por realizar as actualizações na apresentação e no modelo sempre que necessário.

## 2.2.2. Stripes

O Stripes é uma *Framework* desenvolvida em Java, cujo principal objectivo é o mínimo de configuração necessária para a aplicação, aumentando assim a produtividade e a simplicidade no desenvolvimento de aplicações Web [19]. Essa configuração é efectuada num ficheiro em XML (*Web.xml*), onde são registados um *servlet* e um filtro.

Segue o modelo MVC e situa-se nos segmentos do controlo e da vista, interage mas não interfere no modelo, deixando a responsabilidade da persistência para o programador, que assim pode escolher o modelo da base de dados mais conveniente e o mais adequado para o sistema a ser implementado.

O controlador é descrito num filtro configurado com a localização do directório raiz, onde este procura as classes que implementam a interface *ActionBean* que define um controlador da aplicação. Efectua a associação de cada *ActionBean* ao respectivo URL, como ilustra a Figura 2.



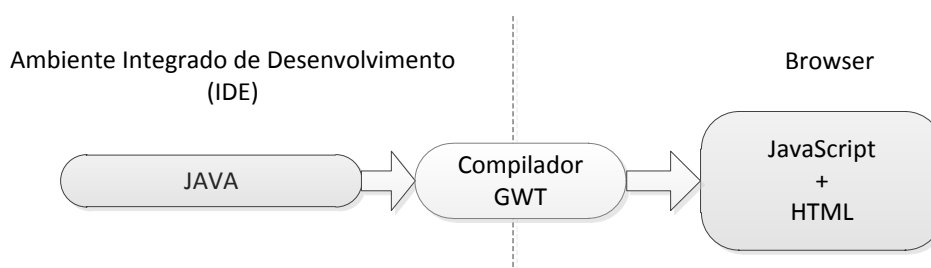
**Figura 2** - Fluxo típico do Stripes [19].

O Stripes utiliza na camada da vista a interface JSP com *tags* próprias e *ActionBeans* referidos anteriormente como controladores, em que a partir do nome completo da classe, o Stripes associa um URL, sendo possível alterar o comportamento usando anotações. Também suporta AJAX através de bibliotecas como o Prototype ou JQuery, permitindo a interacção com a linguagem Javascript.

### 2.2.3. GWT

O GWT (Google Web Toolkit) é um software livre, segundo a licença Apache 2.0 [20], produzido pela Google. É uma plataforma que permite desenvolver aplicações com o recurso à tecnologia AJAX e utilizando a linguagem JavaScript como suporte no lado do cliente. Tem como argumento positivo a facilidade na construção das interfaces, com a possibilidade de *drag-and-drop* na implementação dos componentes GUI e utilização de eventos [21-23].

Permite o desenvolvimento na linguagem de programação Java, e posteriormente efectua a compilação em código Javascript, optimizando e permitindo que seja independente e transparente para os *browsers* Web, como representa o processo na Figura 3.



**Figura 3** - Processo de implementação de uma aplicação GWT.

No desenvolvimento de uma aplicação JAVA em GWT destacam-se duas componentes: o lado cliente, onde o código é compilado em Javascript, e do outro lado, o servidor, em que o código é compilado em Java bytecode.

O GWT fornece o protocolo de comunicação entre o cliente e o servidor utilizando RPC (Remote Procedure Call), em que é definida uma interface que especifica quais os métodos disponíveis. Estando essa interface de utilizador definida, quando um dado

método é solicitado pelo cliente através de um *browser*, o RPC procede à serialização e transferência dos objectos Java de e para o cliente da aplicação. Permite ainda comunicação utilizando tecnologias como JSON ou XML.

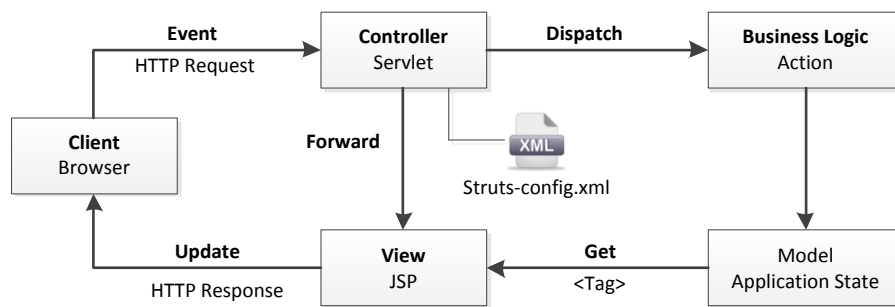
A aplicação pode ser carregada de uma vez só, sendo toda a interacção realizada sem a necessidade de novos pedidos ao servidor. O GWT permite ainda a manipulação da estrutura HTML da página (DOM), usando JavaScript para processar dinamicamente uma página. Permite ainda adicionar e ajustar os vários componentes, o que permite que a aplicação possa actualizar o DOM e responder com os resultados face a eventos de interacção. Quando os dados necessitam de ser enviados ou obtidos a partir do servidor, as aplicações GWT comunicam de forma assíncrona, ou seja, sem interromper as interacções do utilizador, característica da utilização do AJAX.

## 2.2.4. Struts

A *Framework* Struts é um projecto em código aberto com ligação à fundação Apache. Proporciona uma quantidade de objectos que podem facilitar os aspectos fundamentais do MVC, permitindo ainda de acordo com as necessidades que vão surgindo, configurar os objectos base da arquitectura através de ficheiros XML, impondo a utilização de um conjunto de regras bem definidas [24].

A arquitectura MVC é implementada no Struts através de um *servlet* que controla a aplicação, denominado *ActionServlet*, sendo este o responsável pelo controle do fluxo entre as páginas JSP e as diversas camadas da aplicação. Utiliza o *ActionForwards* e *ActionMappings* para manter as decisões do fluxo de controlo fora da camada da vista. Os objectos *ActionMapping* são carregados em memória a partir de um arquivo XML chamado *struts-config.xml*. A representação do fluxo típico no Struts é exemplificada na Figura 4.



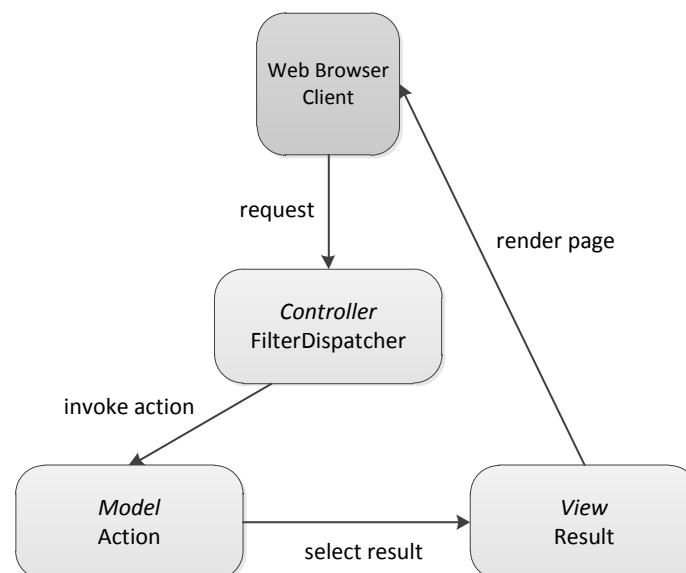


**Figura 4** - Fluxo típico do Struts.

## 2.2.5. Struts 2

O Struts 2 é uma evolução do Struts, baseado no OpenSymphony WebWork [25], e surgiu da necessidade de evoluir e responder às limitações do Struts.

A configuração é efectuada através de anotações, reduzindo as configurações necessárias em XML que eram necessárias na versão anterior. É baseado em *Actions*, suportando componentes modificáveis e reutilizáveis para a interface do utilizador e integra como funcionalidades AJAX nativas, importantes para a validação no lado do cliente, reduzindo a carga no lado do servidor. O fluxo típico do Struts 2 é representado na Figura 5.



**Figura 5** - Fluxo típico do Struts 2 [26].

## 2.2.6. JSF

O JSF (JavaServer Faces) é uma *Framework* que segue o modelo MVC e tem como principal objectivo ser fácil de utilizar, permitindo uma programação simples e por módulos. Fornece um conjunto de ferramentas para que o desenvolvimento de aplicações se torne mais simples, minimizando a quantidade de código necessário. Gera uma maior quantidade de processamento no lado do servidor, reduzindo assim, o código JavaScript no lado do cliente [27].

Possui um conjunto predefinido de componentes para a interface de utilizador, permitindo o desenvolvimento de componentes adicionais através de *tags* personalizáveis, e segue o modelo de programação orientada a eventos. Um ponto negativo a ressaltar é não ter embutido a tecnologia AJAX, muito útil para as aplicações Web da actualidade.

No JSF, o controlo é composto por um *servlet* denominado *FacesServlet*, utilizando arquivos específicos de configuração em XML, e utiliza gestores de acções e eventos [28]. Contém *tags* específicas que são incorporadas no JSP, como se pode verificar através da representação do topo da arquitectura na Figura 6. Foi projectado para ser flexível, não se limitando a uma linguagem particular, protocolo ou dispositivo cliente.

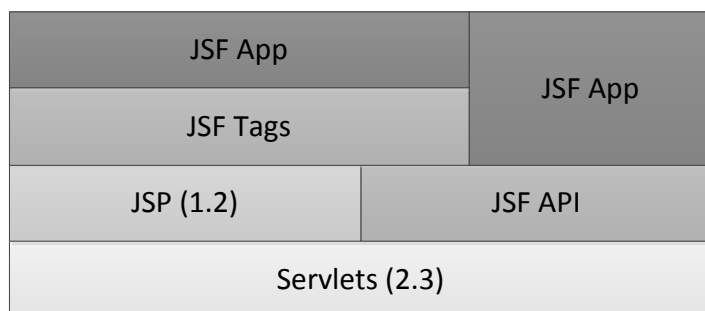


Figura 6 - Topo da arquitectura JSF [28].

## 2.2.7. Análise comparativa

Neste tópico é apresentado o seguinte estudo comparativo, ilustrado na Tabela 1, com vista a escolher a melhor ferramenta para o desenvolvimento da aplicação. Os aspectos importantes a ter em consideração foram a documentação e suporte

disponível, necessidade de configuração, validação de erros, a sua flexibilidade e as ferramentas de apoio para a implementação de segurança.

Para a avaliação dos atributos considerados, é utilizada uma escala de 1 a 3, em que a classificação atribuída a cada um dos valores da escala é a seguinte:

- Mau (1),
- Razoável (2),
- Bom (3).

	Suporte	Aprendizagem	Configuração	Validação	Escalabilidade	Segurança
<b>Stripes</b>	1	3	3	2	3	2
<b>GWT</b>	3	2	3	3	2	2
<b>Struts</b>	2	1	2	2	1	2
<b>Struts 2</b>	2	2	3	2	3	2
<b>JSF</b>	2	2	2	2	2	2

**Tabela 1** - Comparação entre as *Frameworks*.

Em termos de documentação disponível, todas elas possuem bons tutoriais e exemplos de iniciação. Já quanto ao suporte encontra-se um ponto de fragilidade no Stripes, devido à comunidade de utilizadores ser reduzida. Quanto à aprendizagem, apenas de referir que o Struts apresenta mais dificuldade devido à nomenclatura por vezes não ser clara nos componentes que pertencem à arquitectura. Também de notar a demora na compilação no GWT.

No que respeita ao nível de complexidade da configuração, o Stripes leva vantagem, devido à necessidade de uma reduzida e simples configuração, o que não acontece em Struts e em JSF que pelo contrário apresentam aqui pontos negativos. No Struts2 continuam a ser necessários diversos ficheiros de configuração.

Todas estas ferramentas fornecem mecanismos de validação de campos em formulários. JSF, Struts e Stripes efectuem essa operação no lado do servidor, enquanto o GWT efectua essa verificação no lado do cliente. Não se pode estabelecer pontos fortes ou fracos neste tema porque, por um lado vamos sobrecarregar o servidor da aplicação, no outro caso é sobrecarregado no lado do cliente, ou seja, depende da finalidade da aplicação a construir. De notar, que todos apresentam

formas de implementação de segurança, sendo possível desenvolver o mínimo de segurança exigido.

A escolha recaiu na *Framework Stripes*, por nos parecer a mais indicada para desenvolver este trabalho, no que respeita à relação tempo necessário para a implementação do sistema e a facilidade de iniciação na utilização desta ferramenta. Apresenta, como grande vantagem a definição de uma simples configuração inicial e posteriormente não é necessário efectuar mais alterações neste domínio.

## 2.2.8. Outras tecnologias utilizadas

No desenvolvimento deste trabalho foram utilizadas outras tecnologias como o JavaScript, AJAX, a biblioteca jQuery e jQueryUI para melhorar o desempenho e a interactividade da aplicação.

Uma das vantagens do JavaScript é a sua forma simples de permitir aceder e manipular o conteúdo HTML duma página, dando-lhe maior dinâmica [29, 30]. Apesar da sua sintaxe ser idêntica à do Java, nada tem a ver com ela, nem em conceito nem na forma de utilização. Entre um vasto leque de características, o Javascript é uma linguagem interpretada, tendo a possibilidade de carregamento em ficheiros com extensão js. Não existe a necessidade de definir o tipo das variáveis, e contém um bom suporte quanto a utilização das expressões regulares.

Trata-se de uma linguagem de programação do lado do cliente, porque é o *browser* que suporta a carga de processamento. Permite a programação orientada a objectos, com funções, estruturas de dados complexas, colocando à disposição do programador todos os elementos que formam a página Web, para que este possa acedê-los e modificá-los dinamicamente. A grande vantagem de se utilizar JavaScript numa página Web está na possibilidade de realizar diversas tarefas ainda no cliente, antes da informação ser enviada para o servidor, poupando tempo e tráfego de dados.

Quanto à tecnologia AJAX a sua utilização visa diminuir a latência na resposta entre o cliente e servidor numa aplicação Web, tornando a transferência de dados mais eficiente e permitindo ao utilizador que a interacção com a aplicação possa ser flexível e contínua [31-33]. O termo AJAX é utilizado para especificar um conjunto de tecnologias que ajudam as aplicações Web a ficarem mais dinâmicas, evitando o

carregamento de toda página, só actualizando o que for realmente necessário de forma assíncrona. Melhora significativamente a rapidez de resposta, não sendo necessária a actualização ou redireccionamento para uma nova página HTML. Os componentes principais que constituem o AJAX são o XHTML, o modelo DOM, Javascript, XML e CSS.

O jQuery é uma biblioteca JavaScript que além de muito poderosa e fácil de utilizar, consegue abstrair o programador Web do trabalho exigido para a criação de recursos de interactividade numa interface cliente, simplificando e reduzindo o código necessário. É transparente na utilização das aplicações para os diferentes *browsers* [34, 35].

A biblioteca jQuery oferece uma camada de abstracção de propósito geral para criação de scripts em páginas Web. Permite assim serem desenvolvidos, criados e adicionados plugins, com o intuito de adicionar novas funcionalidades. É útil para simplificar a utilização do AJAX.

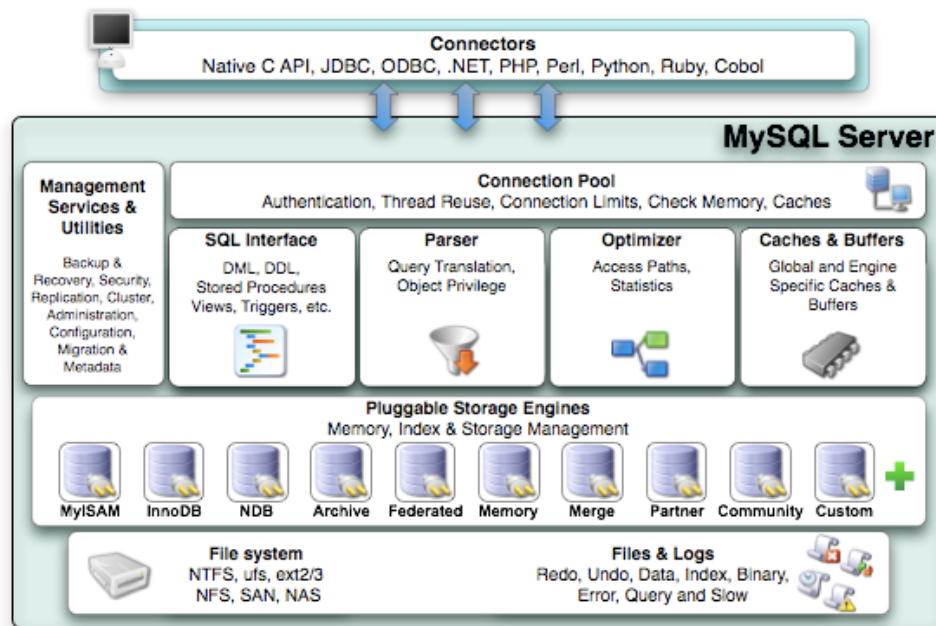
Como extensão ao jQuery surge o jQueryUI que proporciona novas opções para o desenvolvimento de interfaces Web, e tem como objectivo a abstracção na construção de interacções, animações, contendo temas personalizáveis.

## 2.2.9. Persistência de dados

Um dos aspectos mais importantes na implementação de uma aplicação é a persistência dos dados, ou seja, permitir guardar e manipular a informação. Actualmente existem muitas opções em código aberto disponíveis, mas devido à utilização do MySQL por parte da plataforma LOVD que iremos usar no sistema, recaiu aqui a escolha.

O MySQL é um SGBD (Sistema de Gestão de Base de Dados), que utiliza a linguagem SQL e tem como principais características a portabilidade, ou seja, é independente da plataforma em que é integrada, é compatível com diversas linguagens de programação, contém um bom desempenho e estabilidade, sendo pouco exigente em termos de recursos de hardware. Cada ligação cliente corresponde a uma thread

dentro do processo do servidor, não sendo necessário criar ou destruir cada thread numa nova ligação [36]. A arquitectura é ilustrada de acordo com a Figura 7.



**Figura 7** - Arquitectura do mecanismo de armazenamento do MySQL [37].

É necessário, no entanto, a utilização do EclipseLink como suporte para a utilização da *Framework* JPA. É uma biblioteca baseada no TopLink desenvolvida pela Oracle. O principal objectivo e vantagens da utilização do JPA são permitir a abstracção da implementação do modelo de classes, bem como a configuração mapeada na base de dados relacional, permitindo que seja alterada sem ser necessário alterar o código da aplicação. O mapeamento entre os objectos Java e a base de dados é definido no ficheiro *persistence.xml*. No sistema a implementar a configuração efectuada neste ficheiro foi o seguinte:

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.0">
  <persistence-unit name="RDPortalPU" transaction-type="RESOURCE_LOCAL">
    <provider>org.eclipse.persistence.jpa.PersistenceProvider</provider>
    <exclude-unlisted-classes>false</exclude-unlisted-classes>
    <properties>
      <property name="javax.persistence.jdbc.url"
value="jdbc:mysql://localhost:3306/portal"/>
      <property name="javax.persistence.jdbc.password" value="*****"/>
      <property name="javax.persistence.jdbc.driver" value="com.mysql.jdbc.Driver"/>
      <property name="javax.persistence.jdbc.user" value="root"/>
    </properties>
  </persistence-unit>
</persistence>
```

As operações de persistência são geridas e executadas por um *EntityManager*, utilizando métodos como *persist*, *remove* ou *update* de objectos e permite a recolha de informação através de *queries*. As entidades são classes Java, utilizando classes do tipo *EntityBean*. São adicionadas anotações que representam certas características que são reproduzidas na entidade relacional.

## 2.3. Sumário

Neste capítulo foi efectuado uma descrição e caracterização geral do que é uma doença rara e o levantamento do que tem sido feito últimos tempos para melhor a qualidade nesta área da medicina. Foi apresentado também, uma descrição de algumas ferramentas para desenvolvimento de aplicações Web onde foi efectuado um estudo comparativo e a justificada a escolha do Stripes relativamente a pontos que achamos relevantes para este sistema.

## 3. Análise de Requisitos

Como primeiro passo de qualquer processo de desenvolvimento de software é necessário efectuar uma boa análise de requisitos, estabelecendo um conjunto de objectivos gerais que o sistema de cumprir. Neste capítulo são descritas as funcionalidades que foram identificadas como necessárias a desenvolver, de uma forma detalhada.

### 3.1. Visão geral do sistema

A aplicação tem como principal requisito ser *web-based*, ou seja, um servidor que disponibiliza o serviço, permitindo o acesso através de uma ligação HTTP utilizando a Internet. Pretende-se com este sistema, a formação e suporte de uma rede de centros de referência. Estes centros vão ser utilizados por diversos utilizadores com distintos tipos de acesso.

Um centro de referência é um espaço virtual, formado por uma equipa multidisciplinar em que cada membro é responsável por tarefas específicas. Este centro tem utilizadores habilitados para efectuar o rastreio, tratamento e monitorização de um paciente com doenças raras. Os pacientes são acompanhados das mais diversas formas por estes utilizadores (clínicos, laboratórios, geneticistas, ...). Cada centro é geralmente especializado num tipo de enfermidade de forma a concentrarem os melhores especialistas nesse domínio.

Pretende-se que o sistema permita uma configuração inicial do centro de referência, em que serão escolhidas as doenças raras alvo dos estudos a realizar, bem como os formulários personalizáveis, que vão permitir efectuar um registo clínico mais adaptado às circunstâncias pretendidas, e os membros que fazem parte, com diferentes tipos de permissões de acesso.

A configuração inicial de um centro de referência é um processo fundamental no sistema. Quando um centro de referência é criado, é também atribuída a um utilizador a responsabilidade de administração, sendo necessário disponibilizar uma sequência de tarefas para proceder à configuração inicial desse centro.

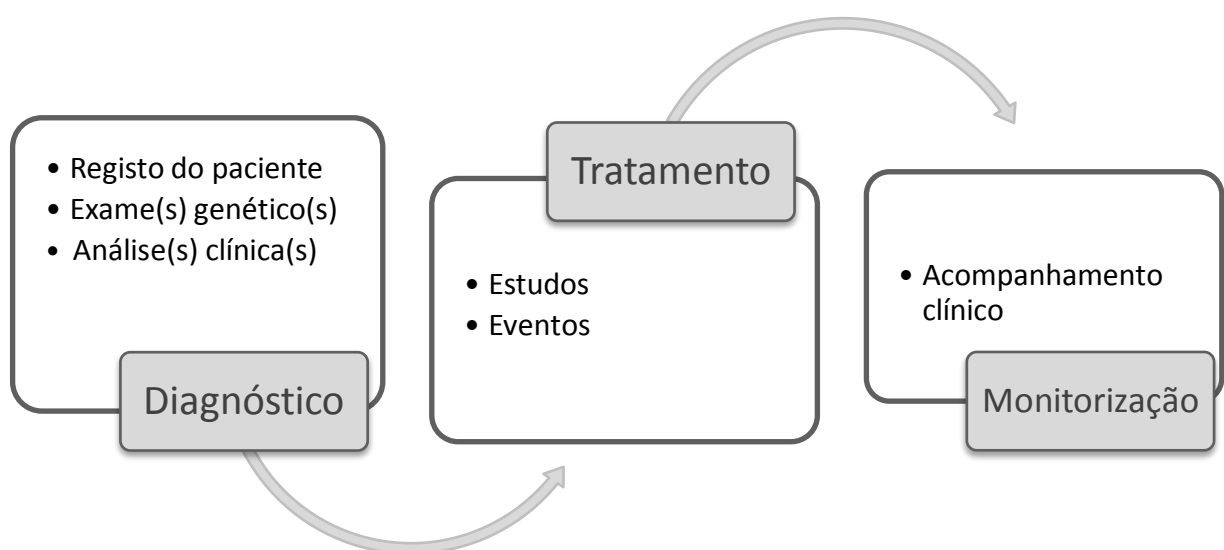


A recolha de informação deve ser efectuada sobre dados disponibilizados pela OMIM e HGNC, como objectivo de elaborar a lista de doenças raras e os genes associados que vão ser alvo de estudo e o desenvolvimento de formulários personalizáveis de acordo com as opções e informação a considerar.

A informação de uma doença rara é obtida pelo ficheiro *morbidmap* da OMIM e a informação dos genes recorrendo à HGNC, efectuando posteriormente um cruzamento dessa informação e disponibilizar os genes que cada doença está associada. É essencial efectuar o tratamento destes dados e fazer um agrupamento das doenças, para que, se reduzam os erros, e sejam associados correctamente os genes dessas doenças raras com os genes apresentados no HGNC. De notar, que existem inconsistências na informação do ficheiro *morbidmap*, algo que será visto com particular atenção na implementação.

Cada paciente do sistema só poderá ter um registo de identificação, possibilitando associar ao longo do tempo os estudos efectuados, realizando todo o registo histórico e permitindo a sua posterior visualização.

O acesso à informação dos pacientes são restritivas, ou seja, independentemente de cada centro de referência a integridade da informação de cada paciente deve ser respeitada e protegida. A estes pacientes, vão ser associados estudos, eventos, variações de genes encontrados na realização de exames genéticos, doença rara associada, evolução, tratamento e monitorização ( Figura 8).



**Figura 8** - Processo do acompanhamento clínico efectuado num paciente.

A estratégia quanto ao registo das mutações genéticas, passa por integrar a aplicação LOVD para os registos genéticos, permitindo assim guardar informação sobre as variações de cada gene relativo a uma doença rara. É essencial criar uma camada superior que permita o registo das mutações genéticas e associá-las aos pacientes, de maneira que possibilite uma identificação e possível diagnóstico de pacientes que chegam ao rastreio ou às consultas. Na realização desses exames genéticos o paciente já deverá estar registado, caso contrário tem de se proceder a esse registo, de modo a que o registo da mutação seja válido.

Quanto às interfaces de utilização a aplicação deverá disponibilizar duas: uma para a gestão da aplicação, utilizada por administradores do sistema e outra como plataforma para os Centros de Referência, utilizada pelos seus membros, que a partir de uma lista de permissões associadas aos mesmos vai dar acesso a determinadas funcionalidade da aplicação. Isto permite que, o portal tenha vários níveis de acesso aos dados de forma a conseguirmos restringir os acessos aos módulos da aplicação.

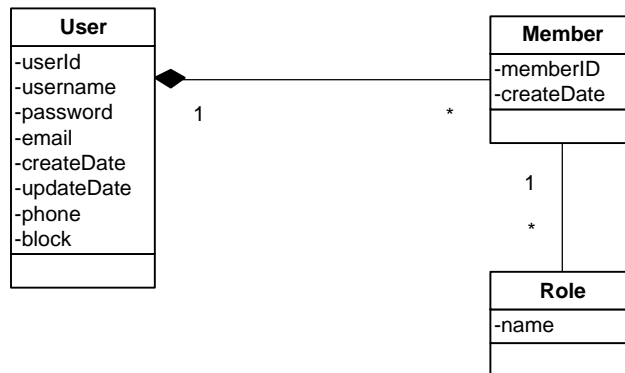
## 3.2. Casos de Utilização

A representação e descrição dos casos de utilização visam compreender as funcionalidades pretendidas, bem como o relacionamento com os actores do sistema. É efectuada de seguida a apresentação dos actores e são caracterizados os casos de utilização para a administração e manutenção do sistema e para gestão e utilização das funcionalidades do centro de referência.

### 3.2.1. Actores

Na aplicação existem dois tipos de actores: o administrador do sistema e o utilizador dos centros de referência. O administrador é o elemento responsável pela aplicação, efectuando a gestão de contas dos utilizadores e, verificando as operações efectuadas nos centros de referência. Tem também como tarefa a criação de novos centro de referência e porventura o envio de convites para novos utilizadores.

Os utilizadores dos centros de referência seguem o modelo de dados ilustrado na Figura 9, em que são associados como membros aos centros de referência e contêm uma lista de permissões de acesso atribuídas.



**Figura 9** - Modelo de dados dos utilizadores.

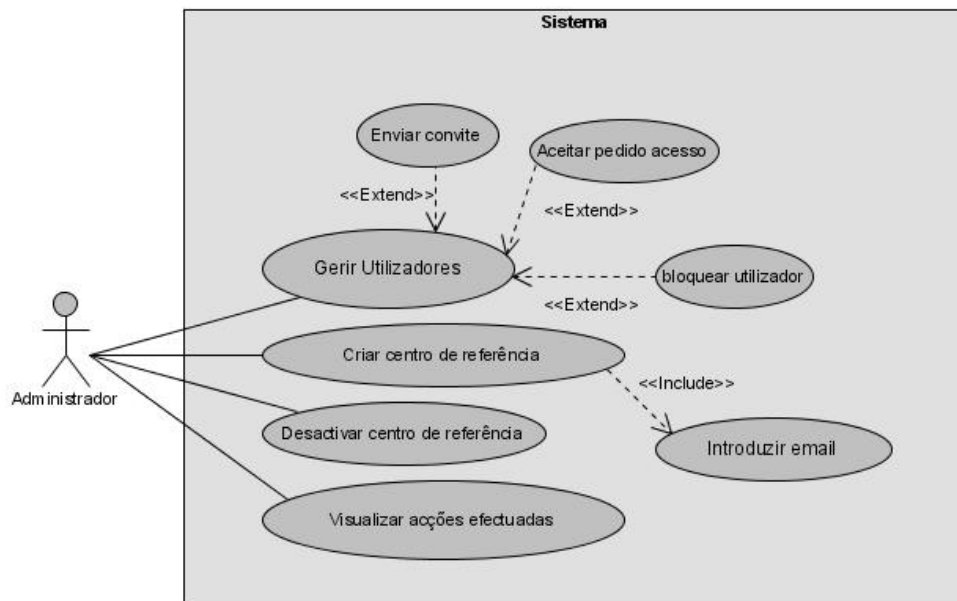
Os actores da aplicação vão ter resumidamente os seguintes papéis no sistema, enumerados na Tabela 2:

Actor	Papel no sistema
Administrator	Gere os registos dos utilizadores Responsável pela criação dos centros de referência
Utilizador	Gestão do centro de referência Efectua o registo do paciente Regista os estudos realizados ao paciente Utilizador dos formulários para o registo clínico Registo das mutações genéticas

**Tabela 2** - Actores do sistema.

### 3.2.2. Administração do sistema

Pretende-se descrever os casos de utilização do administrador do sistema, sendo permitidas as operações ilustradas na Figura 10. Foram considerados os seguintes casos de utilização:



**Figura 10** - Diagrama dos casos de utilização do Administrador.

- **Criar centro de referência:** o administrador pretende iniciar a constituição de um novo centro de referência, introduzindo um nome para o centro de referência e uma breve descrição do mesmo. Delega através do correio electrónico a responsabilidade da administração para um utilizador, que será o responsável pela gestão do centro.
- **Desactivar centro de referência:** não permite mais operações e acesso ao centro de referência.
- **Gerir utilizadores:** é disponibilizada a lista de utilizadores presentes no sistema e permite aceitar um pedido de acesso a um novo utilizador, enviar convites para possíveis utilizadores, ou bloquear um utilizador do sistema, para que não lhe seja permitido entrar na aplicação.
- **Visualizar acções:** permite o acesso à lista de acções efectuadas na aplicação, como estudos efectuados ou registo de pacientes efectuados pelos membros do sistema.

Na Tabela 3 são apresentados resumidamente outros casos de utilização permitidos para a gestão do sistema.

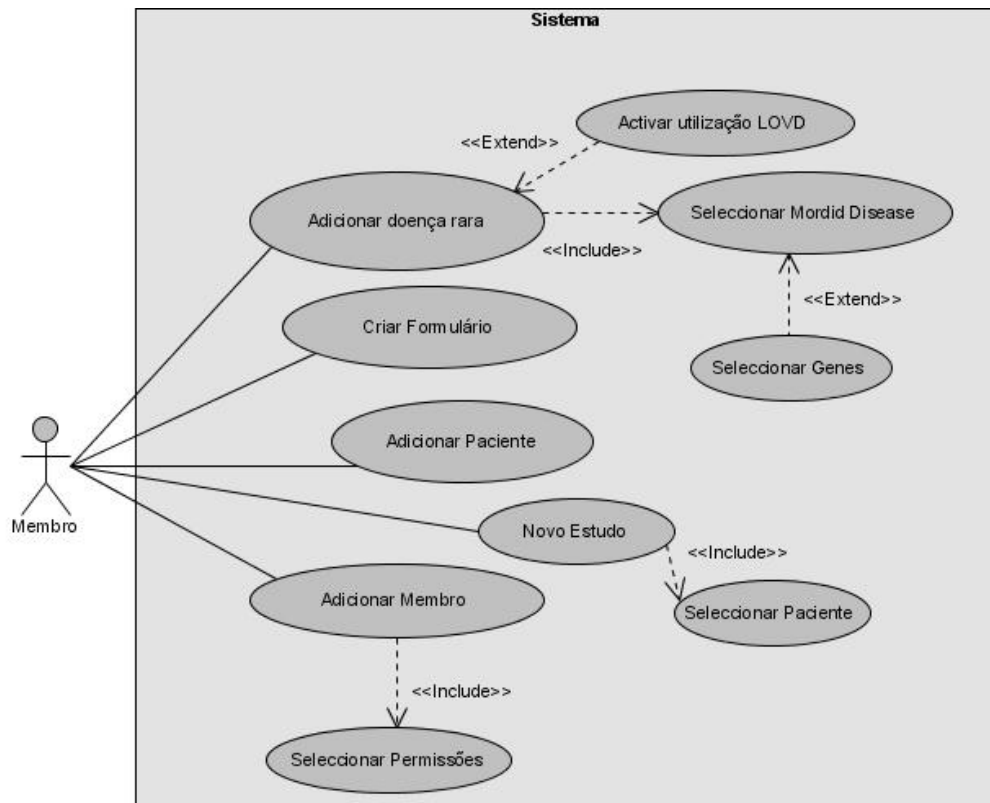
Caso de utilização	Propósito
Bloquear acesso de um utilizador	Não permitir o acesso do utilizador ao sistema.
Aceitar pedido de acesso	Aceitar pedido efectuado para o registo de um utilizador.
Enviar pedido de acesso	Registo de pedido para ingressar no sistema.
Login	Através do fornecimento de credenciais é efectuada a operação de autenticação no sistema

**Tabela 3** - Outros casos de utilização.

### 3.2.3. Administração do centro de referência

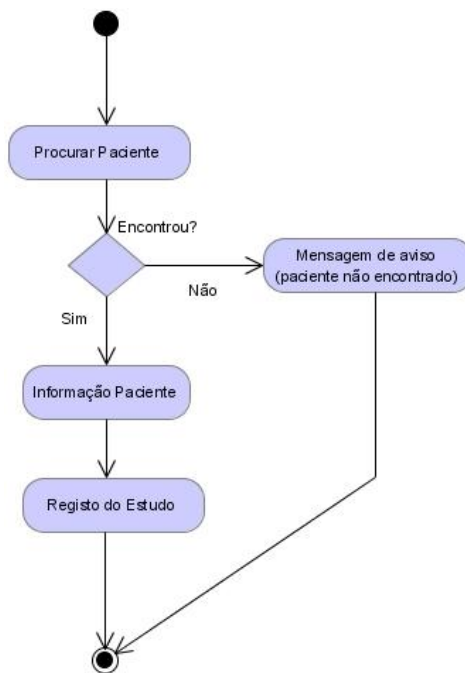
Um membro do centro de referência tem associados os seguintes casos de utilização:

- **Adicionar doença rara:** é apresentada uma lista com as doenças obtidas através de uma recolha de informação na base de dados e possibilita a associação de genes a essa doença sejam inseridos na instância do LOVD.
- **Criar formulário:** apresenta uma lista de opções disponíveis para campos prováveis para integrar o formulário. Permite ao utilizador, de uma forma iterativa e simples, desenvolver uma solução personalizável para ser usada posteriormente nos estudos.



**Figura 11** - Diagrama dos casos de utilização do Membro.

- **Adicionar membro:** é apresentado um formulário que solicita o correio electrónico do novo membro, e permite seleccionar as permissões que este vai conter no centro de referência.
- **Gerir Membros:** é apresentada uma lista com os membros do centro de referência e permite, se o membro tiver a permissão *Admin*, efectuar a alteração e até bloquear um membro.
- **Adicionar paciente:** o utilizador tem acesso a um formulário para proceder ao registo do paciente no sistema.
- **Novo estudo:** permite efectuar o registo clínico de um paciente, como representado no diagrama de actividades da Figura 12. É apresentada a lista de formulários disponíveis para utilização, que, quando usados, são associados automaticamente ao estudo. Permite também incluir notas, eventos e ficheiros.

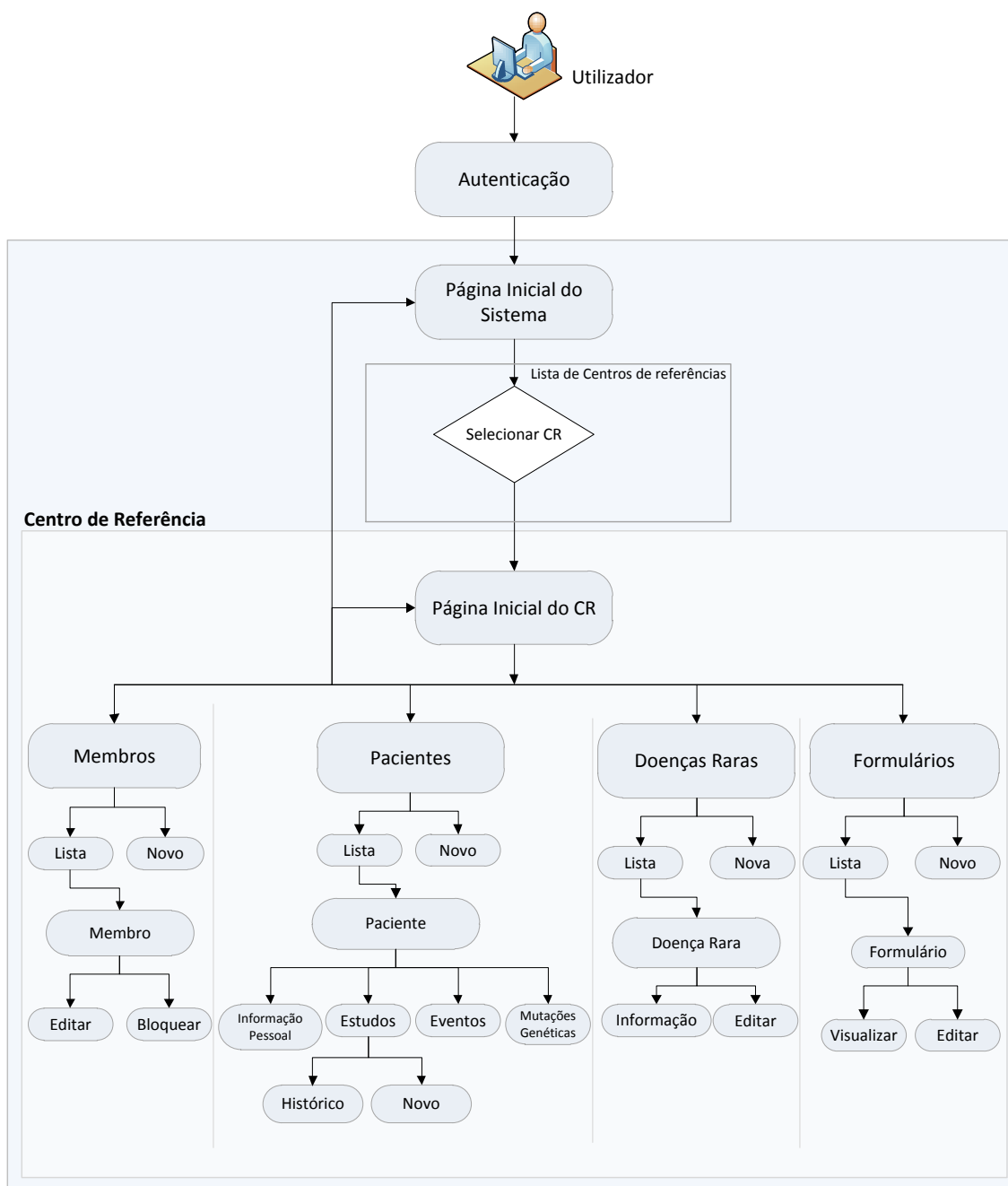


**Figura 12** - Diagrama de actividade de um novo estudo.

- **Registo variação:** permite aceder à interface LOVD e proceder ao registo da variação, seleccionando o paciente que já se encontra disponível na lista de pacientes. Poderá encontrar através da procura pelo ID ou pelo *Nome* do paciente.

### 3.3. Módulos de Trabalho

O sistema contém diferentes tipos de permissão quer na gestão quer na utilização dos recursos disponíveis no funcionamento de um centro de referência. A Figura 13 representa o fluxo de trabalho que o membro tem na aplicação, dependendo no entanto das permissões a ele atribuídas.



**Figura 13** - Diagrama de actividades num centro de referência

As permissões estão divididas do seguinte modo:

- **Membros:** Tem disponíveis operações para a gestão dos membros.
- **Pacientes:** Acesso à informação dos pacientes, possibilitando adicionar estudos.
- **Doenças Raras:** Tem disponíveis operações para a gestão das doenças raras.



- **LOVD:** Acesso à interface LOVD permitindo o registo ou visualização de dados genéticos.
- **Formulários:** Permite a gestão dos formulários utilizados na caracterização das doenças e estudos efectuados aos pacientes.

### 3.4. Requisitos não Funcionais

Para a implementação do sistema, devem ser considerados outros pontos importantes. A plataforma deverá ser de fácil utilização, tendo em consideração o acesso visual às funcionalidades, intuitivo e simples. Devem ser usados ícones de forma a simplificar e tornar mais agradável na sua utilização.

Como preocupação de qualquer aplicação Web, este sistema deve ser compatível com qualquer browser, ou seja, a sua estrutura visual deve-se manter inalterada. Como tal, é um aspecto a ter em conta durante o desenvolvimento.

### 3.5. Sumário

Neste capítulo foram apresentados os requisitos necessários que estão na base do sistema, utilizando diagramas para descrever e identificar o fluxo de informação e operações a disponibilizar. Identificou-se dois actores principais: o Administrador do Sistema e o utilizador, referindo a importância de cada um no sistema. São também identificadas as operações que o sistema deve comportar e que funcionalidades são úteis.

## 4. Desenvolvimento do Sistema de Gestão de Doenças Raras

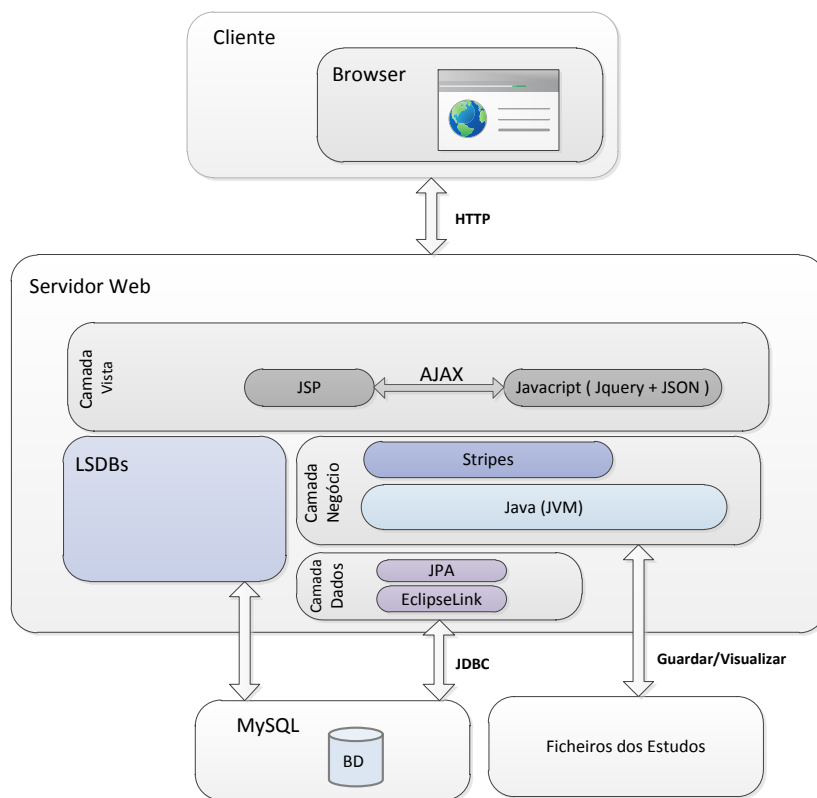
Neste capítulo é descrito o processo de desenho e implementação da arquitectura do sistema, de acordo com os requisitos delineados anteriormente. São especificados as escolhas tomadas em relação a funcionalidades. A implementação da solução é uma fase de desenvolvimento importante porque representa a execução e o aspecto do sistema.

### 4.1. Arquitectura

As decisões arquitecturais são um aspecto fundamental para o sucesso de qualquer aplicação. Numa aplicação Web o componente visual e os respectivos dados encontram-se em espaços físicos diferentes, seguindo deste modo o modelo cliente-servidor, que permite oferecer os serviços com o recurso a uma ligação à Internet. A Figura 14 representa a arquitectura do sistema.

A comunicação entre o cliente e a aplicação é efectuado através do protocolo HTTP. No servidor, para além de se encontrar a aplicação, localizam-se também as LSDBs, que neste sistema são utilizadas instâncias LOVD. Assim, de forma independente, é possível aceder a qualquer interface LOVD sem necessidade de estar autenticado ou registado na aplicação.

Na camada da vista situa-se a interface para o utilizador do sistema. É constituído por páginas JSP e é utilizado JavaScript para a manipulação dos dados no *Browser*. O *Stripes* está presente tanto nesta camada como na camada de negócio, através da utilização de *Urls* e das *ActionsBeans* associadas, utilizando *tags* específicas.



**Figura 14** - Arquitectura geral do sistema.

A camada de negócio, para além da utilização do Stripes com o recurso a ActionBeans contém a interface com a implementação das funcionalidades da aplicação, bem como a ligação e operações para o armazenamento e utilização da informação.

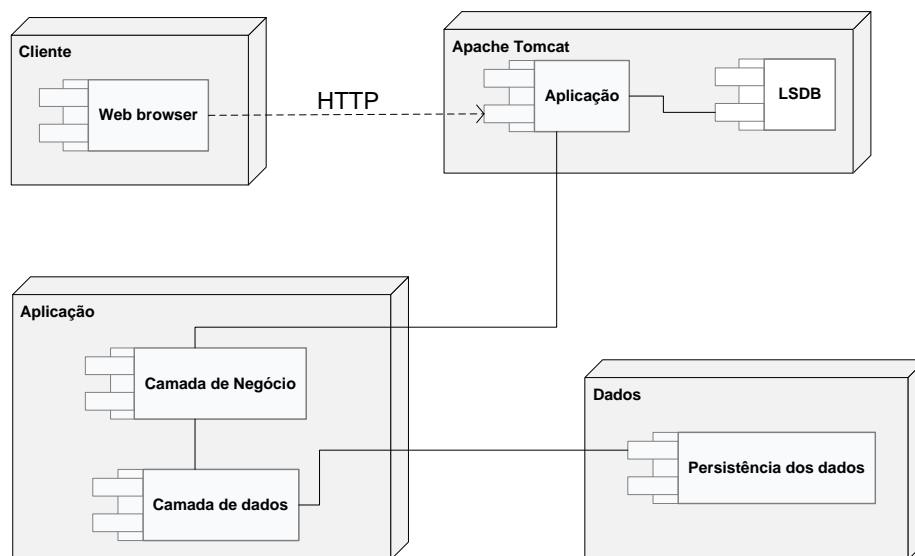
Na camada de dados, a ligação entre os dados e a persistência com o recurso a uma base de dados é utilizado o JDBC. No caso das instâncias LOVD a conectividade com o MySQL é efectuada pelo ficheiro config.ini. Neste segmento são também guardados os ficheiros adicionados ao estudo, através da operação de upload.

#### 4.1.1. Modelo Físico

Esta aplicação segue uma arquitectura cliente-servidor com base em três níveis: apresentação, regras de negócios e persistência dos dados (Figura 15) [38].

Esta arquitectura permite encapsular as operações facilitando o desenvolvimento da aplicação, minimizando a propagação de alterações quando um nível necessita de ser alterado, aumentando a performance, flexibilidade, reutilização.

O nível cliente é responsável por apresentar a informação ao utilizador, de forma a interagir e enviar a informação introduzida até à camada de negócio. Esta camada da arquitectura é a materialização do fluxo de trabalho da aplicação.



**Figura 15** - Modelo físico do sistema.

A camada de negócio implementa os métodos e funcionalidades da aplicação, e interpreta os comandos e dados provenientes do cliente. É a camada principal, porque apresenta grande parte das operações implementadas neste sistema.

A camada de dados é responsável pela transacção e gestão da informação a persistir entre a aplicação e a base de dados. Tem como objectivo, a tradução entre o modelo relativo aos objectos e a persistência da informação, efectuando também a operação inversa.

## 4.2. Modelo de dados

O diagrama de classes UML destina-se a representar o conjunto de classes e respectivas associações que serão implementadas na plataforma. Esta representação serve como suporte para os dados que vão persistir nas tabelas, e que irão compor a base de dados.

### 4.2.1. Centro de Referência

O centro de referência é o elemento central na representação da aplicação na base de dados, pois dele dependem vários objectos. De seguida são descritos em pormenor os conceitos aplicados no desenho da base de dados representados na Figura 16:

- **ReferenceCenter** - Armazena a informação do centro de referência, sendo que cada entrada corresponde a uma instância. Esta tabela pode ser considerada como a raiz do modelo, pois dela depende o registo operacional efectuado no centro. Para cada centro de referência é registada a informação como a data de criação e a *flag build*, booleano útil para se verificar se é necessária a configuração inicial.
- **Form** - Guarda a informação de cada formulário, como o título e descrição que vai permitir identificar o formulário.
- **Field** - Contém a informação do campo associado a um formulário, com dados específicos de modo a permitir que se caracterize de uma forma completa esse campo. Esta informação inclui a posição do campo, se é opcional, os valores por defeito que são incluídos, o tipo de campo e as datas de criação e de edição.
- **Study** - Guarda a informação de um estudo efectuado e que é associado a paciente.
- **RareDisease** - Armazena a informação da doença rara, bem como as recomendações, tratamentos.
- **User** - Contém o registo do utilizador do sistema. Inclui informação como a palavra-chave de autenticação, bem como a informação pessoal.
- **Role** – Guarda a permissões que o sistema suporta.

- **Member** - É a associação entre o utilizador do sistema e o centro de referência, e possui o registo da data de inicio da sua colaboração no centro.

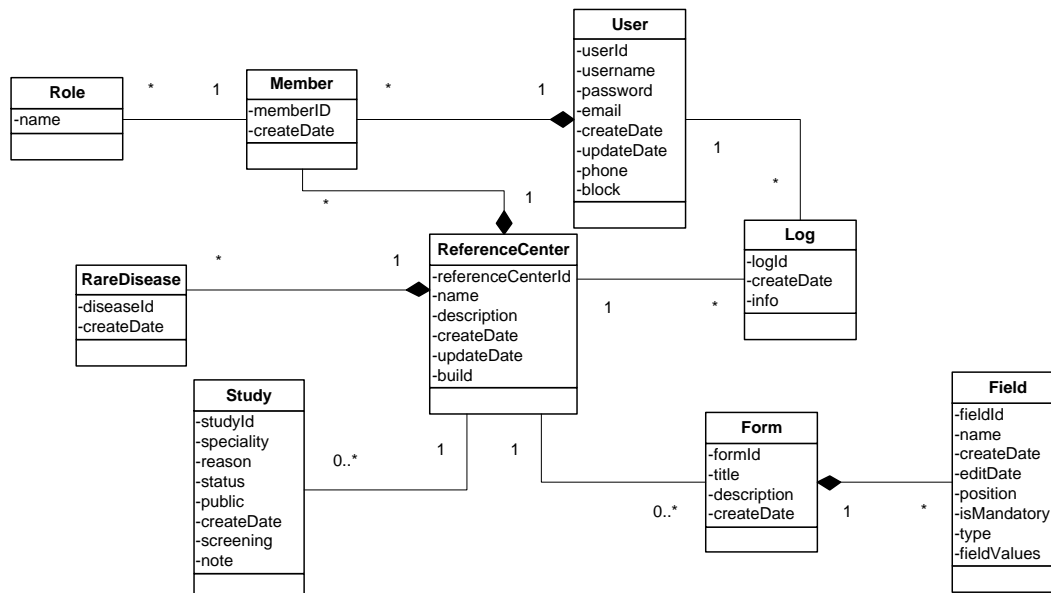


Figura 16 - Diagrama de classes centralizado no centro de referência.

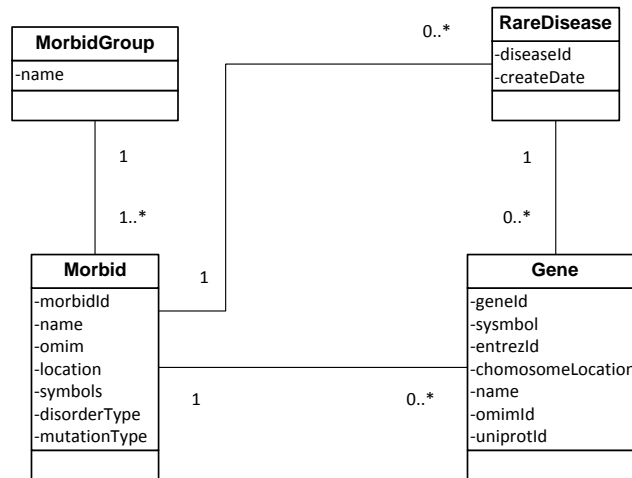
- **Log** - Armazena todas as operações que envolvem o sistema. Inclui informação sobre o nome da operação efectuada e a data que ocorreu.

#### 4.2.2. Informação para as Doenças Raras

A modelação de dados nesta parte específica teve como base a recolha de informação sobre as doenças raras e a posterior utilização por parte dos membros dos centros de referência. A representação do desenho da base de dados é representada na Figura 17:

- **Morbid** - Armazena a informação de uma doença rara recolhida do ficheiro *morbidmap* disponibilizado pela OMIM. Inclui informação como a localização, símbolos identificados, bem como o tipo de mutação.

- **MorbidGroup** – Guarda o nome da família/grupo de cada doença, útil para efectuar uma pesquisa rápida e mais intuitiva.



**Figura 17** - Diagrama de classes centrado na doença rara.

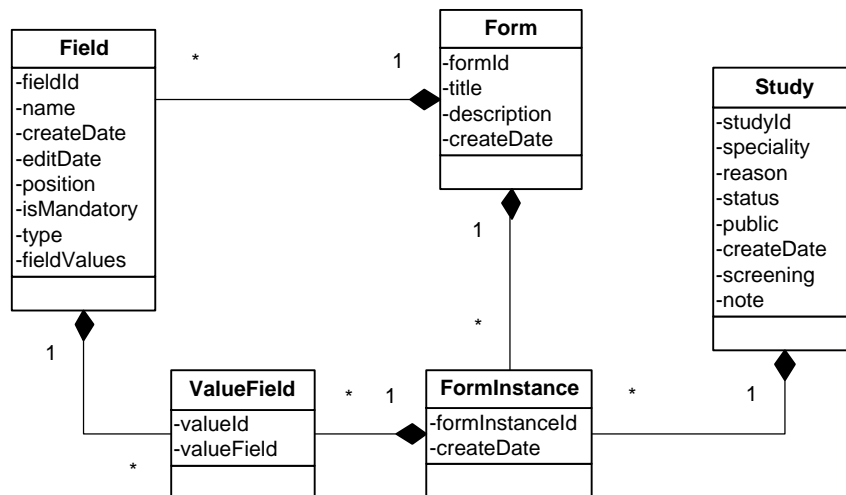
- **Gene** - Contém a informação do gene, associados a uma doença, retirados do ficheiro disponibilizado pelo HGNC. Inclui a informação sobre o número identificador, símbolo e localização do cromossoma. Guarda também os identificadores como os números correspondentes do UniProt, OMIM e Entrez.
- **RareDisease** – Contém a informação sobre a que Morbid corresponde e os genes sobre os quais vão incidir os estudos.

### 4.2.3. Formulário Dinâmico

O modelo de dados escolhido para armazenar a informação de um formulário, deve ter a capacidade de ser dinâmico, ou seja, podendo sofrer alterações ao longo do tempo. A representação dos dados é descrita em pormenor no desenho da base de dados representados na Figura 18:

- **Form** - Guarda a informação de cada formulário. Inclui a informação como o título e descrição que vai permitir identificar o formulário.

- **Field** - Contém a informação do campo associado a um formulário, com dados específicos de modo a permitir que se caracterize de uma forma completa esse campo. Esta informação inclui a posição do campo, se é opcional, os valores por defeito que são incluídos, o tipo de campo e as datas de criação e de edição.



**Figura 18** - Diagrama de classes centralizado no formulário.

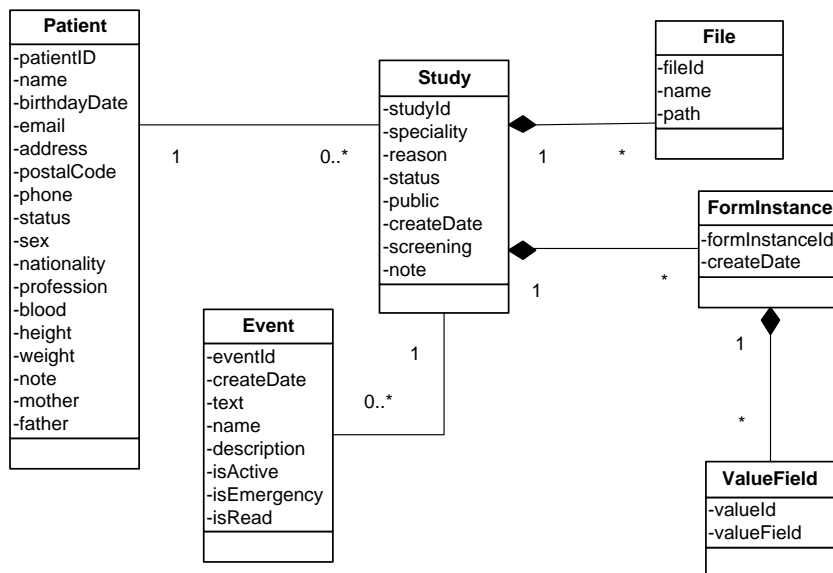
- **FormInstance** - Guarda o registo de uma instância de formulário foi criado. Inclui informação sobre a data que a instância do formulário é criada.
- **ValueField** - Armazena a informação de um campo e estabelece a ligação entre a instância de um formulário utilizado e o campo do formulário modelo.

#### 4.2.4. Estudo

Representa uma parte importante do sistema, onde, através de estes estudos clínicos é possível obter a informação do paciente e a sua evolução ao longo do tempo. De seguida são descritos em pormenor os conceitos aplicados no desenho da base de dados (Figura 19):



- **Study** - Guarda a informação de um estudo efectuado e associado ao paciente. Inclui informação se é público, o motivo, notas e a especialidade do estudo.
- **Event** - Armazena o conteúdo de um evento associado a um estudo. Inclui a data do evento e algumas *flags* como se está activo ou se já foi lido o evento.
- **Patient** - Contém a informação pessoal de um paciente. É criado logo que o paciente entra no sistema, podendo se alterado mais tarde.



**Figura 19** - Diagrama de classes centralizado no estudo.

- **File** - Guarda a informação relativa a um ficheiro associado a um estudo. Contém os campos *nome* e *path*, em que este último refere-se à localização do ficheiro na lista de ficheiros armazenados no modelos de dados do sistema.
- **FormInstance** - Guarda o registo de uma instância de formulário. Inclui informação sobre a data de criação.

## 4.3. Funcionalidades

### 4.3.1. Módulos da Aplicação

A aplicação é dividida em módulos, definidos da seguinte forma:

- Gestão do centro de referência
- Acesso à informação dos pacientes
- Acesso à interface do LOVD
- Acesso a informação das doenças Raras
- Acesso aos formulários

As permissões existentes na aplicação vão ser necessárias para restringir o acesso a determinadas partes da interface. Assim mediante as permissões atribuídas a cada membro, terá acesso autorizado ou negado. A aplicação contém os seguintes cinco níveis de permissões: Admin, Patients, Forms, RareDiseases e Lovd.

**Admin:** acesso a uma barra de ferramentas, situada do lado esquerdo da plataforma. Está habilitado às seguintes operações:

- Adicionar/Editar/Remover Membros
- Adicionar/Editar Doenças raras
- Criar formulários
- Editar formulários
- Adicionar Pacientes
- Ver informação sobre o centro de referência

**Patients:** acesso à informação de um paciente.

- Acesso aos estudos
- Adicionar novo estudo a um paciente
- Acompanhamento do paciente

**Rare Diseases:** habilitado a poder visualizar informação sobre as doenças raras

**Forms:** permissão para acesso no desenvolvimento formulários, editar e visualizar.

**Lovd:** acesso à interface do LOVD, em que poderá efectuar o registo das mutações genéticas que um determinado paciente apresenta.

Para colocar em prática a implementação das permissões foi necessário recorrer ao *plugin Stripes-Security*, que contém uma biblioteca com *tags* de controlo de acesso. É necessário também proceder à indicação no ficheiro de configuração *Web.xml*, registando no *Extension.Packages* o *org.stripesstuff.plugin.security*.

No entanto, é necessária uma classe que efectue a gestão de papéis, a classe *MySecurityActionBean*. Esta classe, estende a implementação do *InstanceBasedSecurityManager*, que por sua vez estende *J2EESecurityManager* e adiciona suporte para restringir as áreas de aplicação de acordo com as roles atribuídas. Isso concede acesso somente se o utilizador tem a *RoleName* papel e de expressão avaliada como verdadeira. A avaliação é efectuada no método *hasRoleName*.

O acesso ao conteúdo é efectuado com o recurso a *tags* que o Stripes tem disponível, podemos ocultar ou apresentar o conteúdo, dependendo da permissão definida. Estas *tags* incluem os seguintes campos de configuração:

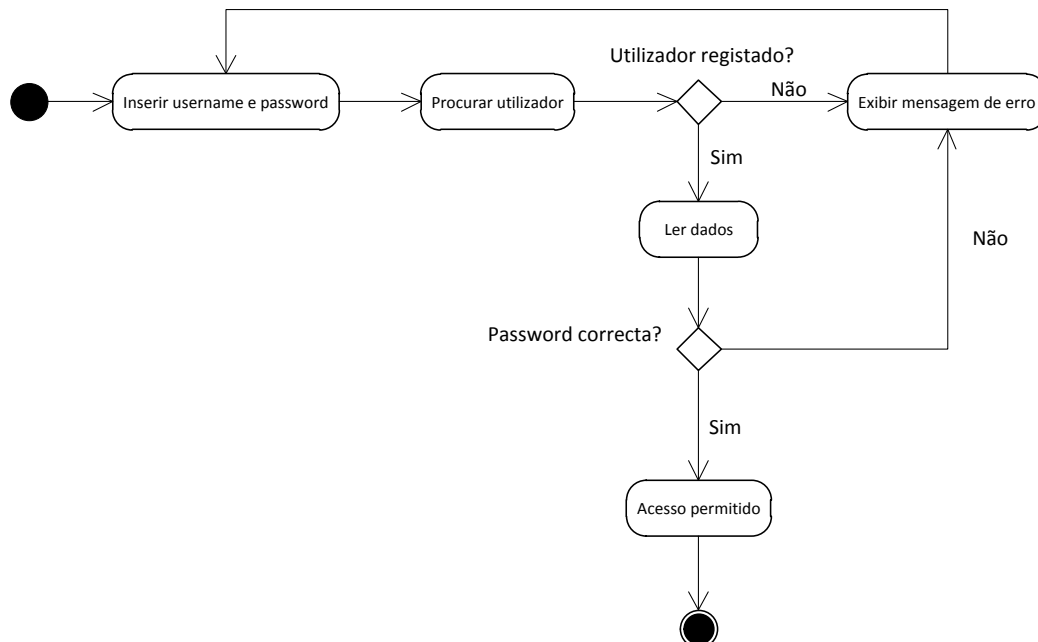
- **id** – representa o identificador da regra.
- **bean** – apresenta o nome da *ActionBean*.
- **event** – representa nome do método associado, a que se refere na *ActionBean*.

### 4.3.2. Autenticação

A autenticação é um aspecto a ter em atenção na aplicação, de modo a garantir autenticidade nas operações e acções efectuadas. Assim, é necessário efectuar com sucesso um login, através de um *username* e *password*, para que o utilizador tenha acesso a interface da aplicação como exemplifica o diagrama de estados representado na Figura 20.

A *password* de acesso é guardada na base de dados utilizando o algoritmo de *hash* MD5, este algoritmo permite grande segurança e a difícil quebra da senha. Assim sempre que o utilizador efectua o login e introduz a *password*, esta é transformada pelo algoritmo MD5 e comparada com os valores existentes na base de dados correspondentes ao seu *username*. Se esta operação for efectuada com sucesso é direccionado para a interface correspondente.

É importante referir, que não é permitido utilizadores com o correio electrónico ou *username* iguais, de forma a garantir unicidade do utilizador registado.



**Figura 20** - Diagrama de estados da autenticação no sistema.

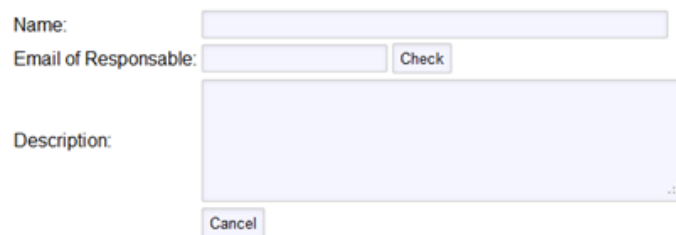
Foi necessário criar variáveis de sessão, de modo a que, durante a sessão do utilizador seja possível guardar valores específicos do estado dessas variáveis. Assim o utilizador poderá navegar nas páginas e essa informação não se perde. Utiliza a interface da API

*Servlet HttpSession*, que contém os métodos *setAttribute* e *getAttribute*, em que é associado a uma chave um objecto [19].

Assim, através da classe acima descrita, foi possível guardar na sessão de cada utilizador a informação sobre o utilizador, a *password* de acesso necessária para a autenticação na aplicação LOVD, informação sobre que centro de referencia se encontra em utilização.

### 4.3.3. Criar Centro de Referência

A plataforma permite que o Administrador do sistema inicie a criação de um projecto, denominado por centro de referência, dando um nome e o correio electrónico do utilizador responsável pelo centro de referência, o formulário de registo é apresentado como se pode verificar na Figura 21.



O formulário para adicionar um novo centro de referência contém os seguintes campos e botões:

- Um campo de texto rotulado "Name:".
- Um campo de texto rotulado "Email of Responsable:" seguido de um botão "Check".
- Um campo de texto de área rotulado "Description:".
- Um botão "Cancel" na base do formulário.

**Figura 21** - Formulário para adicionar um novo centro de referência.

É efectuada a validação dos dados, como o nome do centro de referência que não pode ter o nome de outro já registado e possibilita ao introduzir o correio electrónico do utilizador efectuar uma verificação quanto a existência do correio electrónico e a utilizador corresponde.

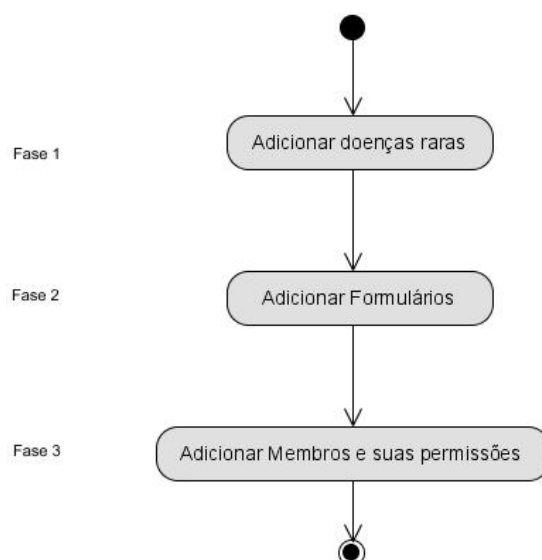
### 4.3.4. Fase de Configuração do Centro de Referência

No início da existência de um centro de referência é necessário colocar à disposição do utilizador funcionalidades para uma configuração inicial. Este processo está dividido em três fases (Figura 22).

Na primeira fase é necessário adicionar as doenças raras para estudo e os genes associados alvo de estudo, recorrendo à possibilidade de criar uma instância LOVD para o registo da informação genética e suas variações.

A segunda fase permite elaborar formulários, que é uma ferramenta que permite personalizar os dados de registo de acordo com as necessidades dos estudos relativo a essa doença. Estes formulários irão ser utilizados pelos médicos durante a fase de rastreio e o acompanhamento as pacientes.

Por último, a terceira fase permite adicionar os membros e as configurações das suas permissões de acesso ao centro de referência.



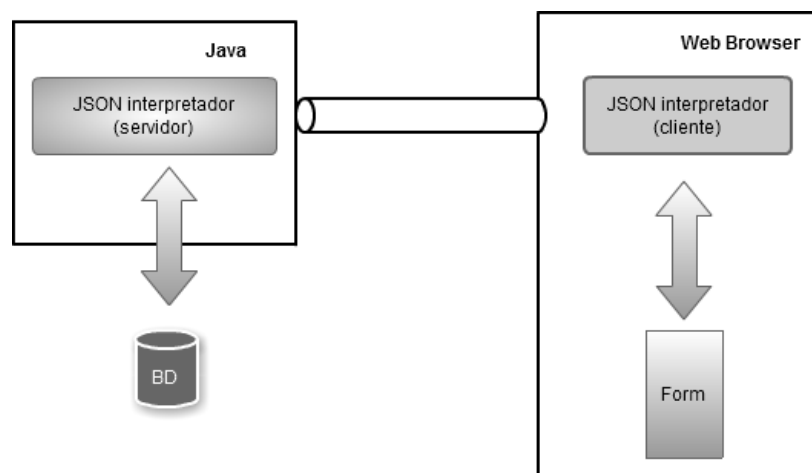
**Figura 22** - Fases de setup de construção de um centro de referência

#### 4.3.5. Formulários Dinâmicos

Este tópico apresenta o processo de modelação e elaboração da ferramenta para a construção dos formulários. É importante realçar, que o objectivo é que os formulários possam ser configurados de acordo com as necessidades do estudo e caracterização de uma determinada doença. Devem permitir também, ao longo do tempo, a possibilidade de alterar a sua estrutura, protegendo a integridade dos dados já existentes.

O formulário contém um título identificador, uma breve descrição e os campos. Os campos podem ser constituídos por diferentes controlos: *textBox*, *multitext*, *selectBox*, *radioButton* ou *checkBox*. Com este cinco tipos de campos pretende-se responder a todas as necessidades de personalização dos formulários.

Este mecanismo suporta a funcionalidade *drag-and-drop*, que torna a construção mais intuitiva e agradável para o utilizador, por forma a facilitar a sua utilização. É possível também alterar a ordem da visualização dos campos previamente adicionados ao formulário. A arquitectura da construção é ilustrada na Figura 23.



**Figura 23** - Arquitectura da construção de um formulário.

Para a formação dos formulários dinâmicos, nomeadamente para representar os objectos formados/criados, é utilizado o formato JSON juntamente com AJAX (através da biblioteca JQuery) que efectua a comunicação entre o cliente e o servidor. Na interface cliente é construído o objecto do formulário, como os campos e os valores por defeito. Concluído este processo, o objecto é enviado para o servidor, que, ao receber a informação, efectua a leitura desse objecto e armazena os dados na base de dados.

Para efectuar a leitura dos dados, como vimos anteriormente, é seguido o mesmo método para a formação dos objectos JSON e a comunicação entre o cliente e servidor. É enviado para o servidor o identificador da instância do formulário, que recolhe a informação e forma os formulários, recorrendo ao formulário modelo que persiste na base dados. Posteriormente esses dados são enviados para a interface cliente sendo composto o formulário e disponibilizado para visualização utilizando JavaScript.

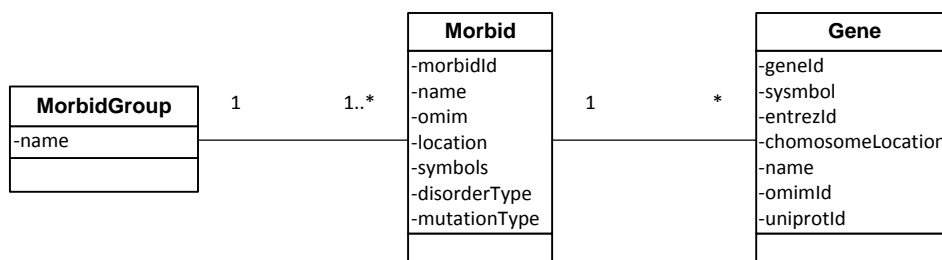
### 4.3.6. Recolha de Informação das Doenças Raras

De modo a permitir uma correcta utilização da informação sobre doenças raras, foi necessário recorrer ao mapa de doenças raras disponibilizado pela OMIM (<ftp://ftp.ncbi.nih.gov/repository/OMIM/ARCHIVE/morbidmap>). Esta informação encontra-se disponível num ficheiro denominado *morbidmap*, já atrás referido, contém para além da informação sobre cada doença rara tem também os genes associados. O formato de cada linha de informação, referente a uma doença rara que se encontra no ficheiro *morbidmap*, é o seguinte:

*Disease title, OMIM id referring to a disease and evidence code | gene symbol and synonyms | OMIM id referring to gene entry | locus*

Os valores estão separados por uma barra vertical. A recolha da informação foi efectuada através com o recurso à biblioteca *opencsv*, um analisador de ficheiros no formato CSV (comma-separated values) e o à utilização de expressões regulares de modo a extrair a informação necessária.

As doenças são guardadas na classe *Morbid*. Posteriormente foi necessário criar uma tabela onde são guardados os grupos das doenças raras – *MorbidGroup* (Figura 24). Esta identificação é importante para aumentar o tempo de resposta quando se efectua a instalação ou na busca de informação. A recolha do grupo é efectuada através do método *split* que separa termos pelo carácter barra e *stopWords* que não queremos que influenciem o nome desse grupo.



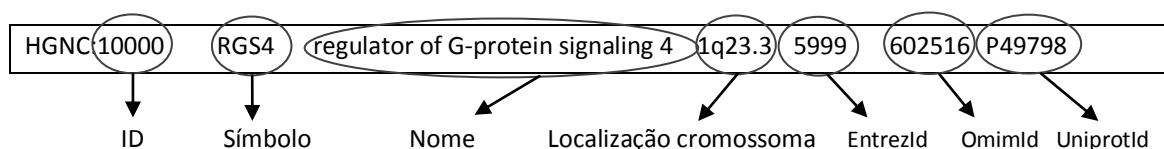
**Figura 24** - Modelo de dados da informação de uma doença rara.

As *stopWords* encontram-se no ficheiro *stopwords.txt* que é lido e carregado num *array* de *strings*. Para cada *token* com o nome da doença rara é verificado se existe



neste *array* de forma a ser ou não adicionado à base de dados. Este processo permite agrupar de uma maneira eficaz o nome do grupo.

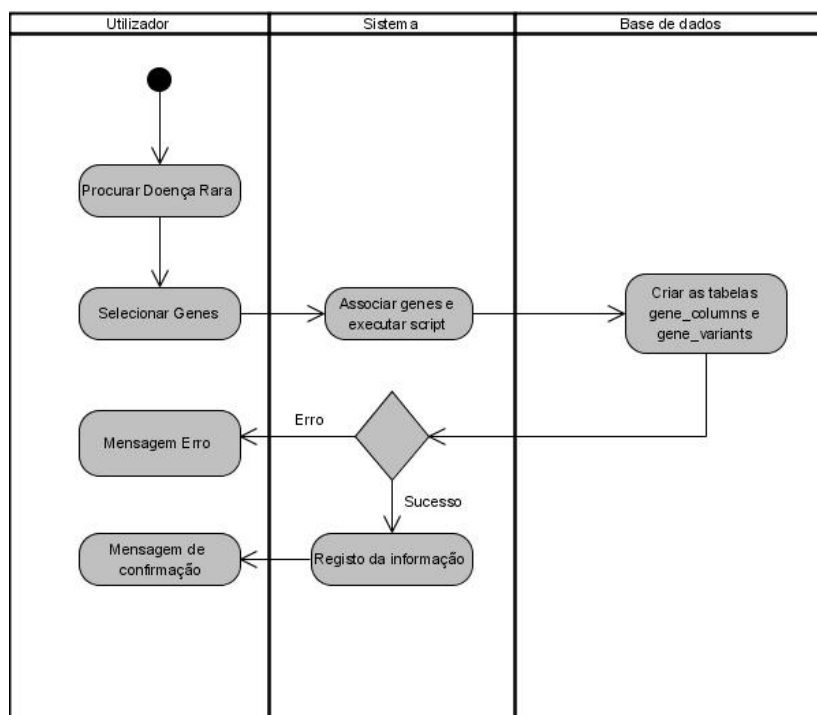
Para obter a informação dos genes relativos a cada entrada do *morbidmap*, encontra-se disponível para download o ficheiro de texto no site do HGNC ([http://www.genenames.org/cgi-bin/hgnc\\_downloads.cgi](http://www.genenames.org/cgi-bin/hgnc_downloads.cgi)), que permite efectuar a correspondência entre os genes e a doença em questão. Assim, quando da fase de instalação é possível escolher os genes alvo dos estudos e que sejam adicionados ao LOVD, permitindo à plataforma ter a disposição toda a informação para proceder à integração com esta aplicação.



Cada entrada contém a informação de um gene, está dividido em sete campos:

- **ID:** identificador único do gene;
- **Símbolo:** símbolo que representa o gene;
- **Nome:** indica o nome do gene;
- **Localização cromossoma:** identifica a localização do cromossoma;
- **EntrezId:** representa o identificador na base de dados Entrez;
- **OmimId:** o número de identificação na base dados OMIM;
- **UniprotId:** indica o valor representado no catálogo apresentado do Uniprot.

Assim, é possível adicionar o gene na base de dados de uma instância LOVD. Para isso foi desenvolvida uma classe com o método *insertGene*, utilizando o prefixo da instância LOVD alvo. Com esse prefixo é criada uma instrução SQL com a informação HGNC recolhida do gene e inserida na base de dados dessa instância. De seguida é necessário criar uma tabela *users2genes*, que identifica o responsável do gene, e adicionar mais duas tabelas à base de dados, *nomeGene\_columns* e *nomeGene\_variants*, que vão conter as variações do genes e as colunas de registo das variações e dos pacientes correspondentes. A Figura 25 apresenta o diagrama de registo de gene no centro de referência e consequentemente na instância LOVD.



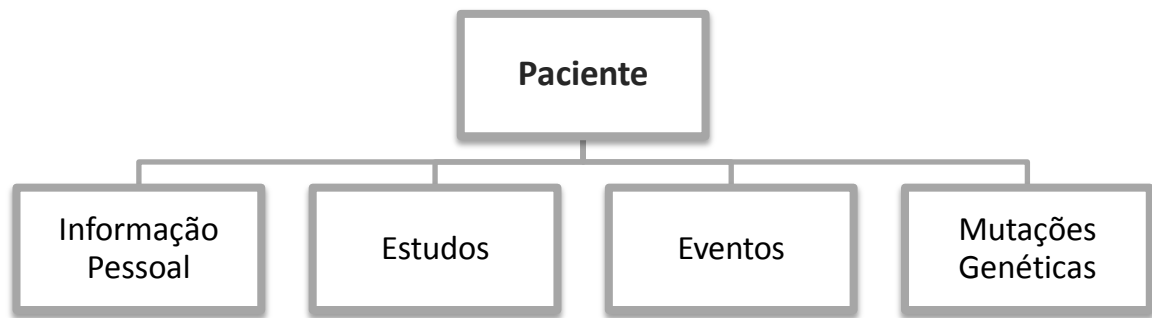
**Figura 25** - Diagrama de sequência do registo dos genes no sistema e no LOVD.

#### 4.3.7. Gestão dos Estudos

Os pacientes constituem uma parte importante da plataforma, uma vez que é sobre eles que são efectuados os estudos, recolhida informação genética, registando o acompanhamento e o tratamento das doenças raras de que padecem.

O registo do paciente é único e só os utilizadores com acesso a esta operação o podem efectuar. Este registo, ao ser único, permite que toda a informação esteja centralizada e disponível para possíveis decisões que o clínico possa efectuar. Também previne a repetição de dados.

A cada paciente são associados os estudos, mutações genéticas, eventos e informação pessoal, como ilustra a Figura 26.



**Figura 26** - Modelo de informação de um paciente.

O sistema disponibiliza uma interface de gestão do paciente contendo os seguintes níveis de representação de informação relativa a um paciente:

- **Informação Pessoal:** Contém informação sobre os dados pessoais do paciente.
- **Estudos:** Apresenta o registo histórico dos estudos efectuados ao paciente, permitindo também efectuar um novo estudo. O utilizador ao efectuar um estudo, tem disponível a utilização dos formulários, bem como efectuar notas, estado, evolução, criar novos eventos ou mesmo juntar algum ficheiro importante do estudo, como imagem médica ou outro tipo de ficheiros úteis. Pode ainda mais tarde, aquando da observação do estudo, efectuar downloads e a visualização dos ficheiros. O sistema guarda na base de dados a localização desse ficheiro, que é armazenado num directório criado para esse estudo.
- **Eventos:** Informação dos eventos associados ao paciente. Podem ser observados em duas perspectivas. A primeira incide nos eventos trocados entre os utilizadores do sistema, para troca de informação, partilha de conhecimento, ou até para administração de consultas. Outra perspectiva a considerar é os eventos criados a partir de um estudo, por exemplo que o paciente deve tomar algum tipo de medicação, ou de alguma outra situação proveniente da execução do estudo.

- **Mutações Genéticas:** Apresenta os registos genéticos encontrados utilizando a interface do LOVD para disponibilizar e apresentar a informação do registo das variações e o gene associado.

### 4.3.8. Registo de acções

Existem dois tipos de registos de acções, uma do lado do Administrador que contém a informação sobre as operações efectuadas no sistema, como:

- Novo utilizador no sistema,
- Novo centro de referência,
- Últimos estudos adicionados no centro de referência.

Esta informação é acompanhada pelo administrador a quem corresponde essa acção, a data da sua criação, bem como a mensagem informativa da operação.

O outro registo de acções prende-se com a actividade de um centro de referência, em que os membros têm acesso à informação dos últimos estudos efectuados nesse centro.

## 4.4. Interacção e Integração com o LOVD

A gestão da informação que o LOVD vai suportar é utilizada como apoio neste sistema, quer no registo de mutações genéticas como na posterior visualização dos resultados, associando esta informação aos pacientes em estudo.

### 4.4.1. Criar uma Instância LOVD

Para a construir uma instância LOVD foi necessário recorrer à utilização de vários scripts. É utilizado um ficheiro em XML chamado *configuration.xml*, que contém a

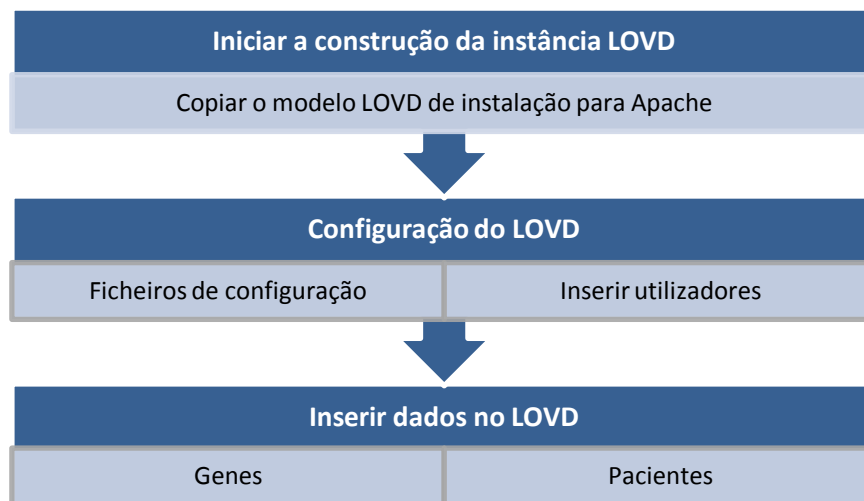
informação relativa à configuração do LOVD informações sobre a instalação, bem como a localização dos ficheiros necessários nesta operação. É ainda necessária a localização do script SQL que descreve a criação das tabelas da base de dados necessária para executar na construção de cada instância do LOVD, bem como a localização destino para onde irão ser criadas essas instâncias.

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <mysql>
    <url>jdbc:mysql://localhost/portal</url>
    <hostname>localhost</hostname>
    <username>root</username>
    <password>*****</password>
    <database>portal</database>
    <engine>InnoDB</engine>
  </mysql>
  <lovd>
    <pathSql>C:/RDPortal/src/java/pt/ua/bioinformatics/RDPortal/scripts/tables.sql
    </pathSql>
    <root>
      C:/Program Files/Apache Software Foundation/Apache2.2/htdocs/Instances/
    </root>
    <copy>c:/lovd</copy>
  </lovd>
</configuration>
```

Foi implementada uma classe chamada *Install* que contém o método *createLovd* cuja função é construir uma instância do LOVD, registando as configurações necessárias para a sua utilização.

O utilizador de acesso ao LOVD será o responsável e administrador do centro de referência. É importante, referir que a *password* e o *username* são os mesmos do sistema de modo a que a utilização do LOVD na plataforma seja transparente, ou seja, que não seja necessário efectuar outro login no LOVD quando estiver a utilizar a aplicação.

É necessário ainda efectuar o registo da informação de configurações, na tabela de configuração da instância LOVD, bem como registar o status do LOVD, nomeadamente a informação relativa à versão do LOVD e nível de actualização efectuadas. O processo de construção é ilustrado na Figura 27.



**Figura 27** - Fases na construção de uma instância LOVD

#### 4.4.2. Gestão das Instâncias LOVD

Cada centro de referência contém um directório único no servidor Web Apache, utilizando o seu identificador da base de dados como nome desse directório, e copiando para esse directório a instância de cada LOVD criado.

Assim, o referido centro de referência contém na sua tabela de instância de LOVD a localização, prefixo, e o nome. De mencionar que as tabelas de cada LOVD estão disponíveis na base de dados da plataforma, permitindo assim uma rápida manipulação dos dados dos dados entre a instância do LOVD e a aplicação.

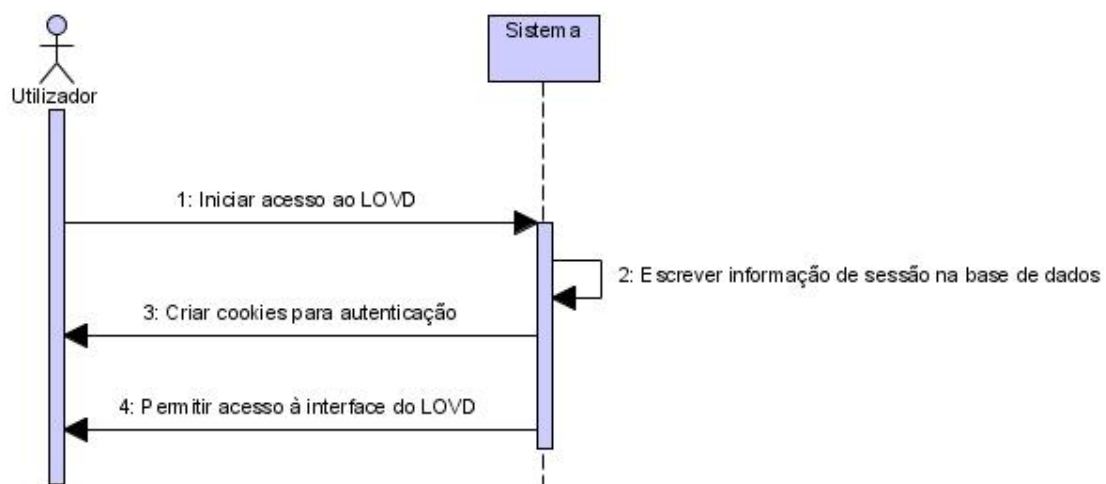
Um dos pontos a favor é que os utilizadores podem ter acesso externamente ao LOVD sem ser necessário entrar na plataforma. Ou seja, pode conter utilizadores que não pertençam a plataforma por exemplo.

A ligação entre a aplicação e o LOVD é efectuada da seguinte maneira: cada centro de referência contém as suas instâncias de LOVD em que são efectuados os registos das mutações genéticas e os pacientes associados. Os utilizadores com acesso ao LOVD têm um registo no LOVD como *submitter*, que, com o recurso a uma autenticação efectuada por cookie poderá executar as tarefas bem como o registo das variações.

Cada instância do LOVD é identificada por um prefixo relativo a um centro de referência. Assim é fácil utilizar as tabelas de cada instância.

### 4.4.3. Autenticação no LOVD

É utilizada uma classe chamada *CookieAuth* cuja funcionalidade é criar os dois cookies necessários para o acesso do utilizador ao LOVD. O primeiro é denominado o *lovd\_cookie\_check* e o segundo é o *PHPSESSID\_localhost* que contém a informação da sessão, bem como a *password* do utilizador. Previamente é alterada a informação na base de dados do LOVD relativa a esse utilizador onde é criada uma *sessionID* com a *password* modificada pelo algoritmo MD5 relativa ao username e password do mesmo.



**Figura 28** - Diagrama de sequência de acesso à interface LOVD.

Como atrás referido a aplicação está dividida em níveis de acesso. Se o membro do centro de referência apresentar a permissão para utilizar o módulo do LOVD é disponibilizado o acesso à interface que liga à instância. Ao seleccionar é apresentada numa iframe embutida na plataforma, podendo o utilizador efectuar as operações que deseja, desde a recolha de informação até o registo de mutações genéticas associadas a um paciente.

#### 4.4.4. Procedimento para Adicionar um Paciente

Na fase de instalação do LOVD, são adicionados à sua tabela *patients* todos os pacientes que se encontram registados na plataforma, associando apenas o seu nome e identificador único. Através deste número identificador do paciente é possível encontrar na instância LOVD a referência para este paciente quando são adicionadas as mutações genéticas associadas.

Para conseguir efectuar estas operações foi necessária alterar a forma da tabela dos pacientes do LOVD. Ficando apenas:

```
CREATE TABLE #prefix#_patients(  
  patientid MEDIUMINT(7) UNSIGNED ZEROFILL NOT NULL AUTO_INCREMENT PRIMARY KEY,  
  Patient/Patient_ID VARCHAR(15) NOT NULL,  
  Patient/Patient_Name VARCHAR(50) NOT NULL,  
  submitterid SMALLINT(5) UNSIGNED ZEROFILL NOT NULL,  
  created_by TINYINT(3) UNSIGNED NOT NULL,  
  created_date DATETIME NOT NULL,  
  edited_by TINYINT(3) UNSIGNED,  
  edited_date DATETIME,  
  INDEX (submitterid))  
TYPE = #engine# ;
```

#### 4.4.5. Recolha de Informação de Mutações Genéticas Relativas a um Paciente

Quando é efectuada uma pesquisa a todas as instâncias do LOVD em que o paciente está incluído, é verificado se este está associado a algum registo de mutação de algum gene previamente adicionado pelo utilizador do LOVD. Essa informação é disponibilizada na apresentação da informação relativa ao paciente na secção de mutações genéticas.



## 4.5. Sumário

Neste capítulo foi discutida a arquitectura do sistema implementada e apresentadas as várias soluções que estão na base do funcionamento deste portal. É descrito o modelo de dados sobre a informação a ser guardada.

Foi dada especial atenção ao modo como o sistema interage com o LOVD, devido ao facto de ser necessário o registo e visualização das mutações. É também descrito o modelo de dados responsável pela construção e manipulação dos formulários dinâmicos.

## 5. Resultados

Neste capítulo são apresentadas as interfaces da solução e os resultados finais obtidos, descritos e exemplificados os processos na utilização das ferramentas implementadas no sistema.

### 5.1. Login e Autenticação

O acesso à aplicação é realizado através de um *Web browser*. Contudo, é necessária a autenticação para permitir o acesso à interface de utilizador. Para tal, é necessário que anteriormente tenha procedido a um registo, utilizando um *username* e *password* próprios. Deste modo, quando se acede ao portal, é apresentada a página Web exemplificada na Figura 29.

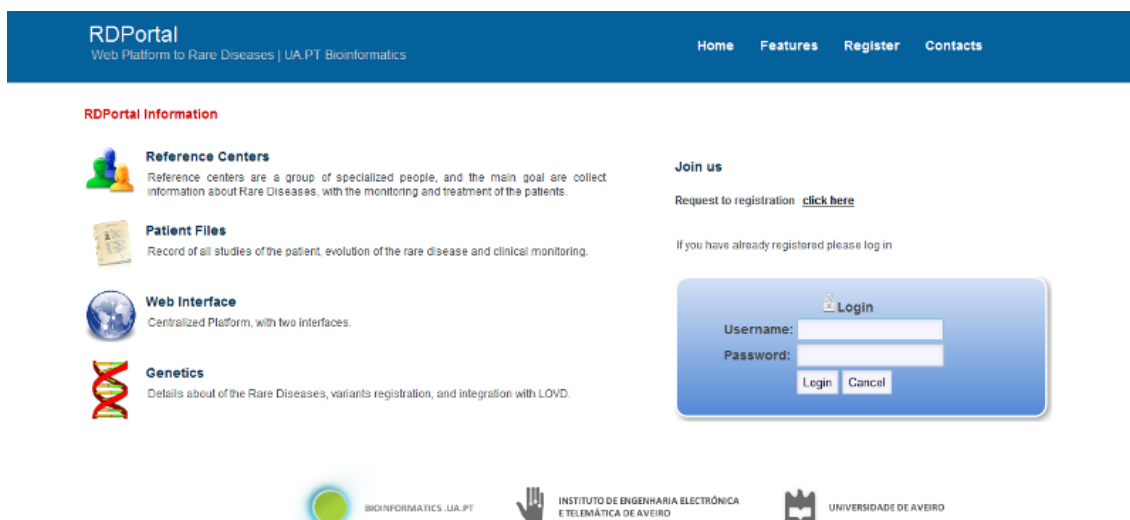
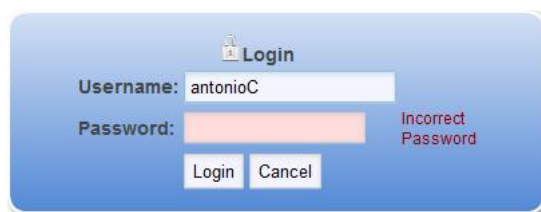


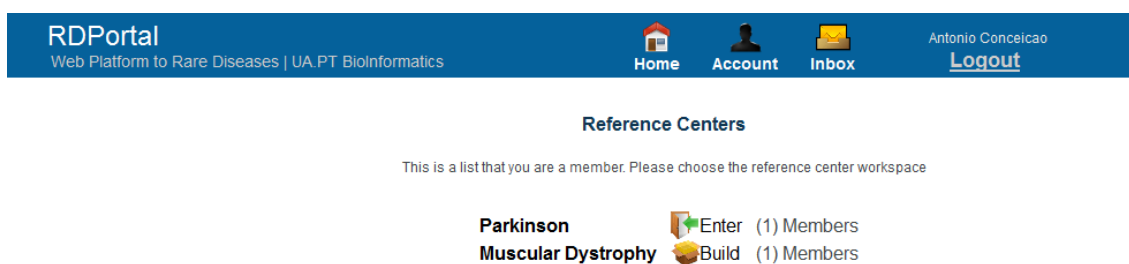
Figura 29 - Página de autenticação.

Ao efectuar o login, os dados colocados no campo *username* e *password* são enviados para o servidor, que ao receber estes argumentos, faz a validação. O resultado produzido por essa validação é mostrado na interface cliente. Consoante o valor retornado, é apresentada uma mensagem de erro caso a autenticação falhe, como é ilustrado na Figura 30, ou no caso de sucesso, tem acesso aos dados e centros de referência associados a este, como podemos verificar na Figura 31.



**Figura 30** - Exemplificação da inserção de dados num login incorrectos.

Ao analisarmos a Figura 31, verifica-se que é disponibilizado ao utilizador a lista de centros de referência de que é membro. Contém uma breve descrição dos centros de referência por exemplo, o número de utilizadores que cada centro de referência tem registado. Se o utilizador é o responsável e ainda for necessária a primeira configuração é apresentado um ícone *build* que permite o acesso à configuração inicial.

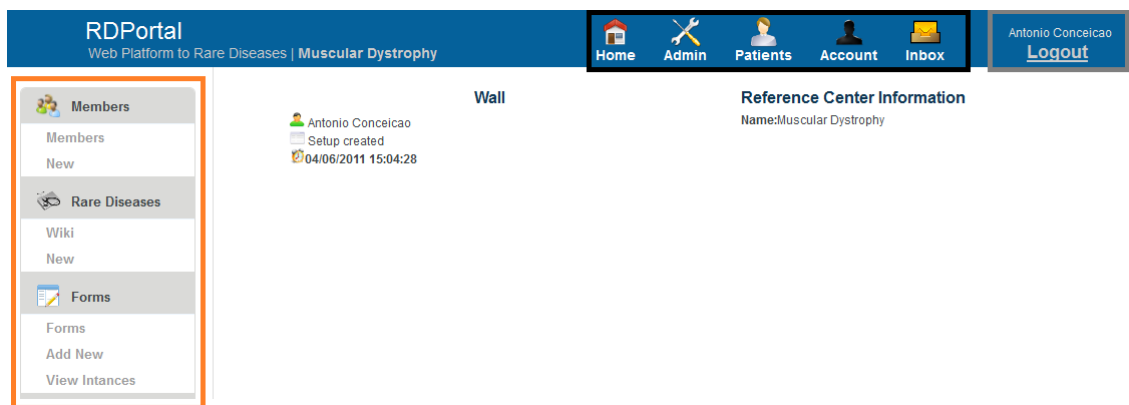


**Figura 31** - Interface cliente após validação do utilizador.

## 5.2. Centro de Referência

A página inicial do centro de referência disponibiliza a informação dos últimos estudos efectuados. Como podemos verificar na Figura 32 contém uma barra lateral, seleccionada com uma caixa laranja, que corresponde às opções disponíveis ao utilizador. Em cima, em preto, representam-se as funcionalidades a que o utilizador tem acesso. Neste caso, tem os botões:

- **Home:** para volta à lista dos centros de referência em que é membro;
- **Admin:** corresponde às opções de administração;
- **Patients:** indica a permissão à informação dos pacientes.
- **Account** - corresponde às informações pessoais e definições de acesso;
- **Inbox:** refere-se à lista de eventos ou mensagens a ele enviado.

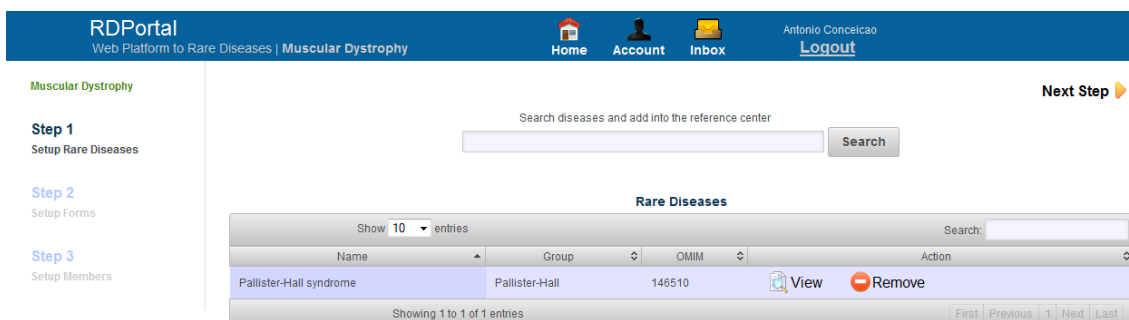


**Figura 32** - Exemplo da página inicial de um centro de referência.

Dentro da caixa em cinzento encontra-se a identificação do utilizador que está com a sessão iniciada com a visualização do nome e permite efectuar o logout e terminar a sessão na aplicação. É apresentado também uma breve descrição do centro de referência, bem como as últimas operações efectuadas nesse centro de referência.

### 5.2.1. Configuração Inicial

Ao iniciar a configuração do centro de referência vai ser apresentada uma interface, como ilustra a Figura 33 quando à importação da informação das doenças raras que vão ser alvo de estudo neste centro.



**Figura 33** - Configuração inicial de um centro de referência.

Contém uma barra lateral à esquerda, que informa o utilizador do passo em que se encontra, permitindo ao utilizador escolher o passo se este não pretender seguir a ordem, já que não interfere em nada o sucesso desta operação.

Search diseases and add into the reference center

**Enable LOVD** ☐

**Name:** Muscular dystrophy with epidermolysis bullosa simplex ☐

**OMIM:** [226670](#)

**Location:** 8q24

**Symbols:** PLEC1, PLEC, PLTN, EBS1, LGMD2Q

**Genes available:**

PLEC plectin ☐

**Name:** Muscular dystrophy, congenital, 1B ☐

**OMIM:**

**Location:** 1q42

**Symbols:** MDC1B

**Genes not found in HGNC**

**Name:** Muscular dystrophy, congenital, due to ITGA7 deficiency ☐

**OMIM:** [613204](#)

**Location:** 12q13

**Symbols:** ITGA7

**Genes available:**

ITGA7 integrin, alpha 7 ☐

**Figura 34** - Visualização da lista de doenças raras.

No primeiro passo da configuração é possível efectuar uma procura, através de grupos de doenças raras, obtido como já referido anteriormente com o recurso ao *morbiditymap*. Contém um sistema *autocomplete* para facilitar as procuras. Ao efectuar uma pesquisa vão ser disponibilizadas as doenças raras associadas a esse grupo, como ilustra a Figura 35. O objectivo é sugerir o nome do grupo de doenças que se encontram na base de dados, precavendo para problemas e erros de sintaxe ou nomes que não existem.

Muscle glycogenosis

Muscle hypertrophy

Muscle strength

Muscular dystrophy

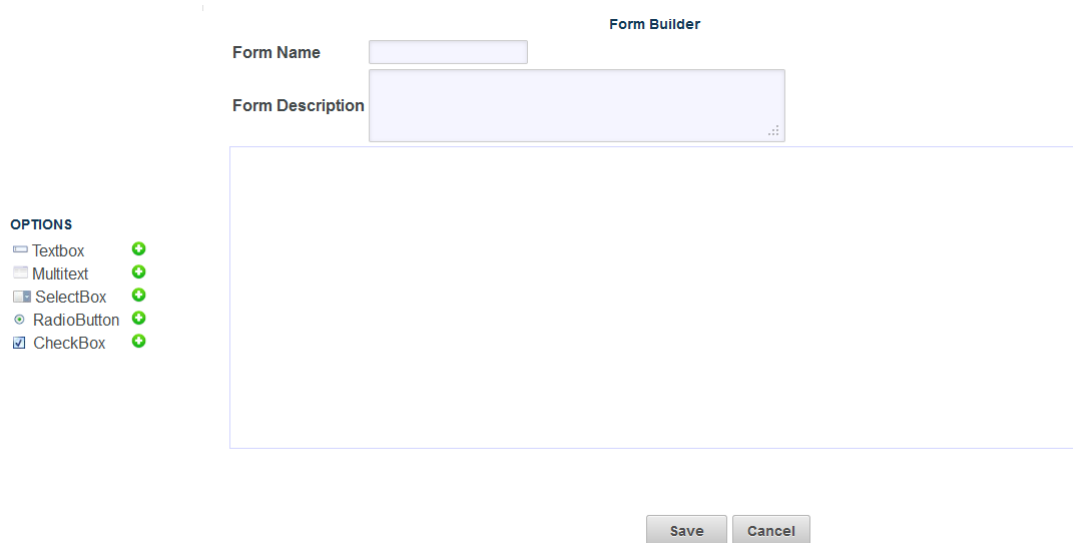
Muscular dystrophy-dystroglycanopathy

**Figura 35** - Exemplo da utilização do *autocomplete*.

Por cada grupo de doenças raras encontradas é possível seleccionar a doença e incluí-la para estudos no centro de referência. Permite o acesso a uma instância LOVD, caso

seja necessário, para seleccionar os genes alvo de estudos. O processo de instalação do LOVD e associação dos genes é transparente ao utilizador e efectuado automaticamente pelo sistema.

Ainda no processo de configuração é possível criar os formulários, que são uma peça fundamental para efectuar os registos clínicos. Foi desenvolvida uma ferramenta que permite de uma forma simples e intuitiva arrastar os campos predefinidos para a elaboração do formulário (Figura 36).

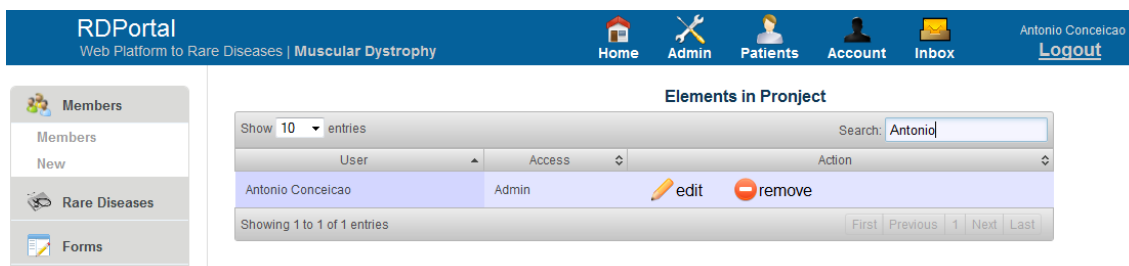


**Figura 36** - Interface para criar um novo formulário.

Do lado esquerdo desta ferramenta está o menu *OPTIONS* com os campos permitidos na configuração e elaboração do formulário. O utilizador pode arrastar para a área de construção e personalizar todos os campos que vão persistir no formulário.

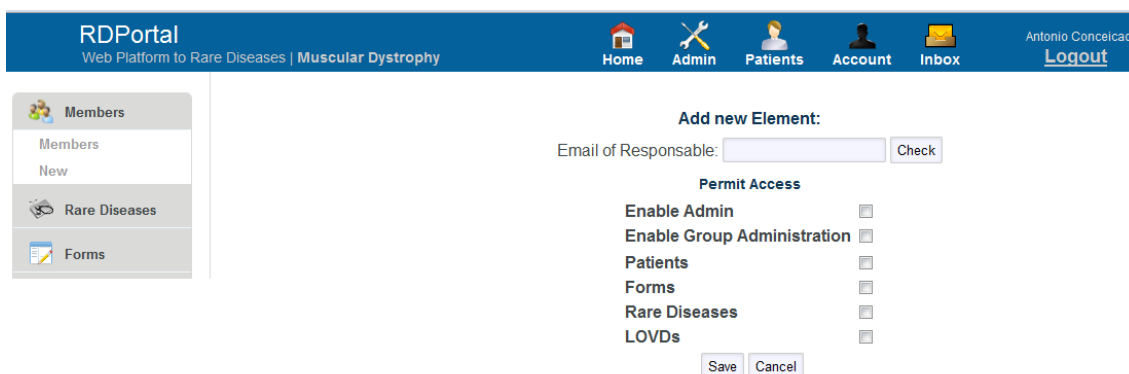
## 5.2.2. Gestão dos Membros

A gestão dos membros constituintes do centro referência é apresentado sobre a forma da interface, ilustrada na Figura 37.



**Figura 37** - Lista de membros que fazem parte do centro de referência.

É apresentada uma lista de utilizadores, permitindo efectuar uma procura pelo nome do membro e tendo a funcionalidade de poder bloquear ou possivelmente editar as permissões de acesso. Outra funcionalidade disponível é permitir adicionar um novo membro ao centro de referência (Figura 38).

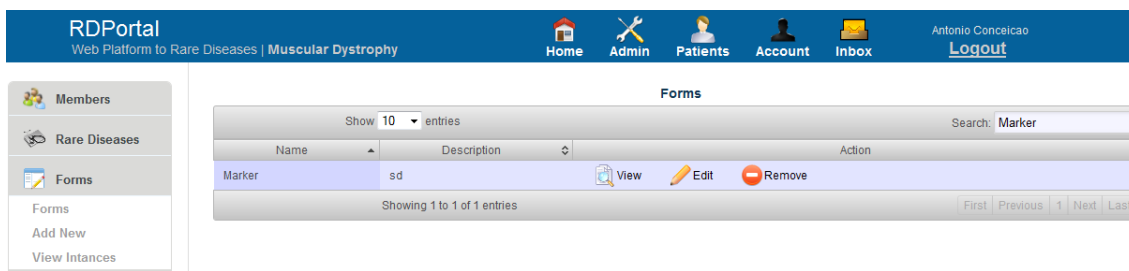


**Figura 38** - Interface para adicionar um novo membro.

No procedimento de adicionar um novo membro ao centro de referência, é necessário introduzir no formulário o correio electrónico do utilizador correspondente, em que o sistema devolve o nome do utilizador para que seja confirmado, e seleccionar as permissões que este vai ter acesso.

### 5.2.3. Gestão dos Formulários Dinâmicos

Quanto ao módulo de gestão de formulários, para além de permitir construir um novo, apresenta as funcionalidades de listar todos os formulários modelo existentes no centro de referência, permitindo visualizar editar ou mesmo apagar, como ilustra a interface da Figura 39.



**Figura 39** - Lista de formulários disponíveis para utilização nos estudos.

É também dada a possibilidade de efectuar a visualização das instâncias de formulários utilizados, útil para a visualização e recolha de dados para ajudar em decisões ou investigação.

#### 5.2.4. Gestão dos Pacientes

A interface de dados sobre a informação dos pacientes é uma peça fundamental neste sistema. É apresentada a lista de pacientes existentes na plataforma, como se pode ver na Figura 40. De referir que só os estudos efectuados nesse mesmo centro de referência é que estão disponíveis para visualização.



**Figura 40** - Lista de pacientes presentes na aplicação.

Contém a funcionalidade de pesquisa através do número de identificação, nome, data de nascimento ou correio electrónico, assim como alterar o número de pacientes por página. É possível também se o paciente não existir no sistema, proceder ao seu registo preenchendo um formulário com as informações pessoais (Figura 41).



**Register new Patient:**

BI:

Name:

Address:

Postal Code:

Email:

Phone Number:

Birthday Date:

Nationality:

Status:

Profession:

Blood group:

Sex:

Weight:

Height:

Mother:

Father:

**Figura 41** - Formulário para o registo de um novo paciente.

O preenchimento do formulário compreende o registo de campos considerados obrigatórios, como o número do bilhete de identidade, o nome e o correio electrónico.

Caso o membro pretenda aceder aos dados do paciente, a informação está dividida em diferentes áreas, tornando mais simples a visualização (Figura 42).

**Vitor Miguel Silva Moreira**

Information

**Patient Information**

Identification: 12440212  
 Name: Vitor Miguel Silva Moreira  
 Address: Paredes, Porto  
 Postal Code: 3880-021 Porto  
 Email: a55555@ua.pt  
 Phone Number: 381222  
 Data de Nascimento: 10/02/2005  
 Status: Single  
 Profession: Student  
 Blood Group: 0+  
 Sex: M  
 Weight:  
 Height:  
 Father:  
 Mother: Eva Cristina Magalhães

**Figura 42** - Interface para a apresentação dos dados relativo a um paciente.

No processo de registo de um estudo efectuado pelo membro do centro de referência, têm à sua disposição as seguintes funcionalidades apresentadas na Figura 43.

**RDPortal**  
Web Platform to Rare Diseases | Muscular Dystrophy

Home Admin Patients Account Inbox Antonio Conceicao Logout

Info  
Forms  
Events  
Upload Files

**Study**

Patient Name: Vitor Miguel Silva Moreira

Reason:

Note:

Screening: ☐

Rare Disease diagnosed: Muscular dystrophy with epidermolysis bullosa simplex

Choose  
Muscular dystrophy with epidermolysis bullosa simplex  
Muscular dystrophy, congenital, due to ITGA7 deficiency

Save Cancel

**Figura 43** - Interface de um estudo.

É apresentado um formulário simples para o registo da informação básica do estudo, como a razão ou notas a considerar, e a doença rara que é alvo do estudo. Permite, como é visível através do menu esquerdo desta interface, a utilização de formulários para o registo clínico, associar eventos a este estudo e o upload de qualquer tipo de ficheiros.

## 5.2.5. Recolha e Registo de Informação de Doenças Raras

No que diz respeito à informação das doenças raras, é utilizada uma lista em que é possível efectuar pesquisas, quer por nome quer pelo identificador da OMIM. Permite para além de adicionar uma nova doença rara na realização dos estudos, excluir ou editar, como é representativo a Figura 44.

**RDPortal**  
Web Platform to Rare Diseases | Muscular Dystrophy

Home Admin Patients Account Inbox Antonio Conceicao Logout

Members  
Rare Diseases  
Wiki  
New  
Forms

**Rare Diseases**

Show 10 entries Search: Muscular

Name	OMIM	Action
Muscular dystrophy with epidermolysis bullosa simplex	226670	View Edit Remove
Muscular dystrophy, congenital, due to ITGA7 deficiency	613204	View Edit Remove

Showing 1 to 2 of 2 entries First Previous 1 Next Last

**Figura 44** - Lista de doenças raras incluídas no centro de referência.

## 5.2.6. Utilização do LOVD

O registo de mutações genéticas tem uma grande importância na colecção de dados relativos a um paciente. Essas mutações podem ser registadas e visualizadas através de uma interface LOVD, como ilustra a Figura 45.

RDPortal  
Web Platform to Rare Diseases | Muscular Dystrophy

Home Admin Patients Rare Diseases Account Inbox Antonio Conceicao Logout

**LOVD** Muscular Dystrophy  
hydroxy-delta-5-steroid dehydrogenase, 3 beta- and steroid delta-isomerase 2 (HSD3B2)  
Curator: Antonio Conceicao

LOVD v.2.0 Build 28 [ Current LOVD status ]  
Welcome, Antonio Conceicao  
Your account | Log out

Home Variants Submitters Submit Configuration Setup Documentation

Submit new data

**LOVD - Submit new sequence variant**

Please note that your submission is only stored *after* you click the button "Finalize submission".

**Patient data**

Patient ID	Name
12440212	Vitor Miguel Silva Moreira

**Sequence variant data**

Add a new variant » Use an existing variant »

Finalize submission

Powered by LOVD v.2.0 Build 28  
©2004-2010 Leiden University Medical Center

Figura 45 - Interface LOVD utilizada pelo sistema.

Para aceder à informação de um paciente ou associar a uma mutação genética apenas é necessário efectuar uma pesquisa introduzindo o número identificador e prosseguir com o registo. Quando efectuado o registo da mutação é apresentado o resultado na área de informação de mutações genéticas do paciente em questão.

## 5.3. Sumário

Neste capítulo foram apresentados os resultados práticos do portal, mostrando a forma como sistema foi implementado e construído, dando maior ênfase às interfaces e funcionalidades do portal.

## 6. Conclusão

Na fase inicial do trabalho foi descrita a importância da arquitectura MVC como elemento fundamental na construção de aplicações Web, e efectuada uma análise comparativa das *Frameworks* para a implementação e desenvolvimento do sistema. Este tipo de tecnologias permitem desenvolver aplicações Web cada vez mais robustas e mais produtivas, obtendo facilidades na acessibilidade e manutenção.

O sistema seleccionado permitiu desenvolver uma plataforma Web que possibilita a gestão e suporte na constituição e na gestão de centros de referência. Apenas com uma ligação à internet os utilizadores têm acesso à informação, de uma forma simples, podendo efectuar a gestão dos estudos clínicos e o registo de mutações genéticas de uma maneira integrada, ligando a informação clínica dos pacientes com os registos das mutações. De salientar também o nível de personalização que se pode obter utilizando os formulários dinâmicos, podendo-se adaptar de acordo com as exigências e objectivos pretendidos.

Esta solução contém um grande potencial de crescimento devido à sua construção por módulos, com a possibilidade de ao longo do tempo, sempre que necessário, acrescentar novas funcionalidade e poder ser adaptado a outras áreas que necessitem de afectação e gestão de recursos. O módulo desenvolvido na construção dos formulários dinâmicos e a possibilidade de interacção e integração com outras LSDBs torna a aplicação flexível.

A aplicação necessita de ganhar maturidade já que ainda não foi colocada em actividade, de forma a avaliar o seu comportamento e o feedback dos utilizadores quanto à satisfação e qualidade do sistema. O desenvolvimento da aplicação revelou-se um grande desafio e a incorporação do LOVD foi bem conseguida porque permitiu melhorar a qualidade no registo e apresentação das mutações genéticas associadas aos pacientes do sistema.

### 6.1. Sugestões para trabalhos futuros

Qualquer trabalho de investigação, bem como o desenvolvimento de uma aplicação, deixa sempre a possibilidade de melhorar ou aumentar as capacidades que o sistema

pode conter. Algumas sugestões de trabalho a efectuar como complemento ao aqui apresentado são:

- Uma interface para que o paciente e familiares possam ser parte integrante deste sistema e tenham acesso a alguns dados úteis.
- Disponibilizar dados estatísticos aos administradores do sistema, sobre as doenças que estão a ser seguidas em cada centro de referência.
- Mecanismos de segurança, importantes para aumentar a confiança dos utilizadores da aplicação.

## 7. Referências

1. *Programa Nacional para Doenças Raras* 2007; Available from: <http://www.portaldasaude.pt/NR/rdonlyres/555DD3B3-45F0-4F74-B633-28889E721BF1/0/i010420.pdf>.
2. Aymé, S. and J. Schmidtke, *Networking for rare diseases: a necessity for Europe*. Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz, 2007. **50**(12): p. 1477-1483.
3. Schieppati, A., et al., *Why rare diseases are an important medical and social issue*. The Lancet, 2008. **371**(9629): p. 2039-2041.
4. Burke, W., *Genetic testing*. N Engl J Med, 2002. **347**(23): p. 1867-75.
5. *An overview of our activities*. 26-05-2011]; Available from: [http://archive.eurordis.org/article.php?id\\_article=1390](http://archive.eurordis.org/article.php?id_article=1390).
6. *EURORDIS Partners in New European Registries Project*. 23 de Maio, 2011]; Available from: <http://www.eurordis.org/content/eurordis-partners-new-european-registries-project>.
7. *Rare Diseases Task Force Mandate*. 2007 04-02-2011]; Available from: <http://www.orpha.net/testor/doc/RDTFmandate.pdf>.
8. *E-rare : Objectives* 26-05-2011]; Available from: <http://www.e-rare.eu/The-Project/Objectives.html>.
9. *About NORD*. 26-05-2011]; Available from: <http://www.rarediseases.org/about>.
10. Hamosh, A., et al., *Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders*. Nucleic acids research, 2005. **33**(suppl 1): p. D514.
11. Povey, S., et al., *The HUGO Gene Nomenclature Committee (HGNC)*. Human Genetics, 2001. **109**(6): p. 678-680.
12. Bruford, E.A., et al., *The HGNC Database in 2008: a resource for the human genome*. Nucleic acids research, 2008. **36**(suppl 1): p. D445.
13. Wain, H.M., et al., *Genew: the human gene nomenclature database*. Nucleic acids research, 2002. **30**(1): p. 169.
14. Claustres, M., et al., *Time for a unified system of mutation description and reporting: a review of locus-specific mutation databases*. Genome Research, 2002. **12**(5): p. 680.
15. *MutaBASE: Background*. 04-07-2011]; Available from: [http://www.mutabase.com/index.php?option=com\\_content&view=article&id=47&Itemid=53](http://www.mutabase.com/index.php?option=com_content&view=article&id=47&Itemid=53).
16. Bérout, C., et al., *UMD (Universal mutation database): a generic software to build and analyze locus-specific databases*. Human mutation, 2000. **15**(1): p. 86-94.
17. Fokkema, I.F.A.C., J.T. den Dunnen, and P.E.M. Taschner, *LOVD: Easy creation of a locus specific sequence variation database using an "LSDB in a box" approach*. Human mutation, 2005. **26**(2): p. 63-68.
18. Singh, I., B. Stearns, and M. Johnson, *Designing enterprise applications with the J2EE platform*. 2002: Addison-Wesley Longman Publishing Co., Inc.
19. Daoud, F., *Stripes: ...and Java web development is fun again*. 2008.
20. *Apache License, Version 2.0*. 01-06-2011]; Available from: <http://www.apache.org/licenses/LICENSE-2.0.html>.

21. Hanson, R. and A. Tacy, *GWT in Action: Easy Ajax with the Google Web Toolkit*. 2007: Manning Publications Co. Greenwich, CT, USA.
22. Chaganti, P., *Google Web Toolkit: GWT Java Ajax Programming*. 2007: Packt.
23. Dewsbury, R., *Google Web Toolkit Applications*. 2007.
24. Cavaness, C., *Programming jakarta struts*. 2004: O'Reilly Media, Inc.
25. Lightbody, P. and J. Carreira, *WebWork in action*. 2006: Dreamtech Press.
26. Brown, D., C.M. Davis, and S. Stanlick, *Struts 2 in action*. 2008: Albazaar.
27. Bergsten, H., *JavaServer faces*. 2004: O'Reilly & Associates, Inc.
28. *JavaServer Faces Technology Overview*. 31-05-2011]; Available from: <http://www.oracle.com/technetwork/java/javaee/overview-140548.html>.
29. Richards, G., et al., *An analysis of the dynamic behavior of JavaScript programs*. SIGPLAN Not., 2010. **45**(6): p. 1-12.
30. Chugh, R., et al., *Staged information flow for javascript*. SIGPLAN Not., 2009. **44**(6): p. 50-62.
31. Dahlan, A.A. and T. Nishimura, *Implementation of asynchronous predictive fetch to improve the performance of Ajax-enabled web applications*, in *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services*. 2008, ACM: Linz, Austria. p. 345-350.
32. Albert, T.J., K. Qian, and X. Fu, *Race condition in Ajax-based web application*, in *Proceedings of the 46th Annual Southeast Regional Conference on XX*. 2008, ACM: Auburn, Alabama. p. 390-393.
33. Fu, Y., et al., *Ajax-based report pages as incrementally rendered views*, in *Proceedings of the 2010 international conference on Management of data*. 2010, ACM: Indianapolis, Indiana, USA. p. 567-578.
34. Chaffer, J. and K. Swedberg, *Learning jQuery 1.3*. Estados Unidos da América: Packt Publishing, 2009.
35. Bibeault, B. and Y. Katz, *jQuery in Action*. 2008.
36. Schwartz, B., et al., *High performance mysql*. 2008: O'Reilly Media, Inc.
37. *Overview of MySQL Storage Engine Architecture*. 01-06-2011]; Available from: <http://dev.mysql.com/doc/refman/5.1/en/pluggable-storage-overview.html>.
38. Lukashenko, R., M. Vilkelis, and A. Anohina. *Deciding on the architecture of the concept map based knowledge assessment system*. 2008: ACM.